Direct Numerical Simulation of Multi-Phase Flows using Extended Discontinuous Galerkin Methods

Zur Erlangung des akademischen Grades Doktor-Ingenieur (Dr.-Ing.)
genehmigte Dissertation von Martin Smuda aus Dortmund
Tag der Einreichung: 30. Juni 2020, Tag der Prüfung: 20. Oktober 2020

1. Gutachten: Prof. Dr.-Ing. M. Oberlack
2. Gutachten: Prof. Dr. rer. nat. D. Bothe
Darmstadt

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Fachbereich Maschinenbau
Fachgebiet für
Strömungsdynamik

Direct Numerical Simulation
of Multi-Phase Flows using Extended Discontinuous Galerkin Methods

genehmigte Dissertation von Martin Smuda

1. Gutachten: Prof. Dr.-Ing. M. Oberlack
2. Gutachten: Prof. Dr. rer. nat. D. Bothe

Tag der Einreichung: 30. Juni 2020
Tag der Prüfung: 20. Oktober 2020

Darmstadt

Kadda

Erklärungen laut Promotionsordnung

§8 Abs. 1 lit. c PromO

Ich versichere hiermit, dass die elektronische Version meiner Dissertation mit der schriftlichen Version übereinstimmt.

§8 Abs. 1 lit. d PromO

Ich versichere hiermit, dass zu einem vorherigen Zeitpunkt noch keine Promotion versucht wurde. In diesem Fall sind nähere Angaben über Zeitpunkt, Hochschule, Dissertationsthema und Ergebnis dieses Versuchs mitzuteilen.

§9 Abs. 1 PromO

Ich versichere hiermit, dass die vorliegende Dissertation selbstständig und nur unter Verwendung der angegebenen Quellen verfasst wurde.

§9 Abs. 2 PromO

Die Arbeit hat bisher noch nicht zu Prüfungszwecken gedient.

Darmstadt, 30. Juni 2020

M. Smuda

Abstract

The scientific study of multi-phase flows is a challenging task for analytical and experimental works. Thus, sophisticated and specialized numerical methods are in need for the direct numerical simulation of such problems.

In this work a high-order multi-phase flow solver on the basis of the extended Discontinuous Galerkin (extended DG/XDG) method is developed. This allows the direct numerical simulation of the transient incompressible two-phase Navier-Stokes equations in their sharp interface formulation. The approximation space of the local ansatz-functions is adapted to be conform to the position of the interface. The interface, described as a level-set function, is discretized by a standard DG method that enables a sub-cell accurate representation of sharp jumps in the pressure field and kinks in the velocity field. For the numerical treatment of the surface tension force the Laplace-Beltrami formulation without regularization is implemented. Stability issues regarding the energy conservation of the solver are addressed.
The developed solver is validated against a wide range of typical two-phase surface tension driven flow phenomena including capillary waves, an oscillating droplet and a rising bubble.

Allowing the simulation of dynamic contact line problems, the generalized Navier boundary condition is adapted for the XDG discretization. The results regarding the rise of liquid in a capillary build the basis of a new benchmark setup for capillarity driven flow problems.

Another extension of the solver is the implementation of the coupled two-phase heat equation in context of the XDG method. Furthermore, the discretization for both the Navier-Stokes equations and the heat equation is extended to allow a mass and energy flow across the interface. This way the velocity field exhibits a sharp jump and the temperature field shows a kink at the interface. A first basic validation is provided against analytical solutions.

This work presents a multi-purpose flow solver for the direct simulation of multi-phase flows involving dynamic contact lines and phase changes due to evaporation. It is based on the XDG method to allow a sub-cell accurate approximation in context of the sharp interface formulation.

Zusammenfassung

Die wissenschaftliche Untersuchung von Mehrphasenströmungen ist eine herausfordernde Aufgabe für analytische und experimentelle Arbeiten. Daher sind hochentwickelte und angepasste numerische Methoden nötig, um derartige Probleme mittels direkter numerischer Simulation zu lösen.

In dieser Arbeit wird ein Mehrphasenströmungslöser auf Grundlage der erweiterten diskontinuierlichen Galerkin (erweiterte DG/XDG) Methode entwickelt, welche ein Verfahren hoher Ordnung darstellt. Dieser Löser erlaubt die direkte numerische Simulation der instationären, inkompressiblen Navier-Stokes Gleichungen, wobei die Phasengrenzfläche eine singuläre Fläche beschreibt. Dabei sind die Ansatzfunktionen zur Position der Phasengrenzfläche angepasst. Die Phasengrenzfläche ist über eine Level-Set Funktion definiert und mittels einer Standard DG-Methode diskretisiert. Dies ermöglicht eine genaue Darstellung von Sprüngen im Druckfeld und Knicken im Geschwindigkeitsfeld innerhalb einer einzelnen Zelle. Zur numerischen Behandlung der Oberflächenkräfte ist die Laplace-Beltrami Formulierung implementiert. Fragen zur Stabilität des Lösers bezüglich der Energieerhaltung werden adressiert.
Der entwickelte Löser wird validiert gegenüber einer breiten Auswahl an typischen zweiphasigen, oberflächenspannungsgetriebenen Phänomenen, wie Kapillarwellen, einen oszillierenden Tropfen und eine aufsteigende Blase.

Die Anpassung der verallgemeinerten Navier-Randbedingung für die XDG-Diskretisierung erlaubt die Simulation von Problemen mit dynamischer Kontaktlinie. Die Ergebnisse zum Aufstieg von einer Flüssigkeit innerhalb einer Kapillare bilden die Basis für einen neuen Benchmark-Testfall für Probleme getrieben durch Kapillareffekte.

Eine zusätzliche Erweiterung des Lösers beschreibt die Implementierung der zweiphasigen Wärmeleitungsgleichung im Kontext der XDG Methode. Weiterhin wird die Diskretisierung der Navier-Stokes Gleichungen und der Wärmeleitungsgleichung erweitert, um einen Massen- und Energiefluss über die Phasengrenzfläche zu erlauben. Dadurch weist das Geschwindigkeitsfeld einen Sprung und das Temperaturfeld einen Knick an der Phasengrenzfläche auf. Eine erste grundlegende Validierung erfolgt anhand analytischer Lösungen.

Diese Arbeit präsentiert einen Mehrzweck-Strömungslöser für die direkte numerische Simulation von Mehrphasenströmungen, welche dynamische Kontaktlinien und einen Phasenwechsel aufgrund von Verdampfung umfassen. Der Löser basiert auf der XDG Methode und erlaubt somit eine akkurate Approximation im Rahmen einer singulären Beschreibung der Phasengrenzfläche.

Acknowledgements

First of all, I want to thank my supervisor Prof. Oberlack for the opportunity and support to write this PhD thesis at the FDY and to give me the opportunity to work in the CRC 1194. Further, I would like to thank Prof. Bothe for the fruitful work within the CRC during the past two years and for being my co-supervisor.

I am deeply thankful to Florian Kummer, who guided me as my first office mate through the BoSSS-code in the beginning of my PhD and for the many discussions in the end during the time in the CRC. Furthermore, I want to express my deepest gratitude to the entire BoSSS-group for the support and proofreading parts of this thesis. Thank you (in alphabetical order) Lauritz Beck, Benjamin Deußen, Markus Geisenhofer, Juan Gutierrez, Dario Klingenberg and Jens Weber. To all colleagues at FDY I would like to thank you for the very nice working atmosphere and countless delicious cakes.

Furthermore, I want to thank some past members of the BoSSS-group, who gave me a lot of helpful input to my work. Thank you Dennis Krause, Stephan Krämer-Eis, Björn Müller and Thomas Utz. A special thanks goes to my friend Anne Kikker for being a supportive and caring office mate, and partner in crime writing our theses.

At last I would like to thank my family and friends for their encouragement and never-ending support. Thanks to my parents, who gave me the opportunity to do my Bachelor and Master studies here in Darmstadt. And the biggest thank you from all of my heart goes to Kadda for her continuous support especially during the last weeks in quarantine with me and this thesis.

This work was to some part supported by the “Excellence Initiative” of the German Federal and State Governments within the Graduate School of Excellence Computational Engineering at Technische Universität Darmstadt and to another part by the German Science Foundation (DFG) within the Collaborative Research Centre 1194 “Interaction between Transport and Wetting Processes.”

Contents

List of Figures

List of Tables

List of symbols

Ω	Domain of interest, computational domain
$\partial\Omega$	Domain boundary
$\mathfrak{A}$	Bulk phase A
$\mathfrak{B}$	Bulk phase B
$\mathfrak{s}$	Bulk phase, species
$\Omega \setminus \mathfrak{I}$	Bulk
$\mathfrak{I}$	Interface
$\mathbf{w}$	Singular surface velocity vector
w	Normal component of the singular surface velocity
$\mathbf{u}_\mathfrak{I}$	Material interface velocity vector
$\mathbf{n}_{\partial\Omega}$	Outer boundary normal vector
$\mathbf{n}_\mathfrak{I}$	Interface normal vector
$\mathbf{n}$	Normal vector
$\boldsymbol{\tau}$	Tangent vector
$\mathbf{P}_\mathfrak{I}$	Interface projection tensor
$\mathbf{P}$	Projection tensor
$\mathbf{u}$	Velocity vector
p	Pressure
e_{kin}	Kinetic energy
e_σ	Surface energy
ψ	General field/physical property
$\mathbf{f}$	Flux vector
ρ	Density
μ	Dynamic viscosity
ν	Kinematic viscosity
σ	Surface tension coefficient
κ	Mean curvature
$\mathbf{D}$	Rate of deformation tensor
$\mathbf{S}$	Stress tensor
θ	Apparent contact angle
θ_{stat}	Static contact angle
L	Contact line
l_{cap}	Capillary length
l_{s}	Slip length
$\mathbf{n}_L$	Contact line normal vector

U_L	Contact line velocity
β	Coefficient of friction
T	Temperature
T_{sat}	Saturation temperature
$\mathbf{q}$	Heat flux vector
c	Specific heat capacity
k	Thermal conductivity
$\dot{m}$	Mass transfer rate
h_{vap}	Enthalpy/(latent) heat of vaporization
A	Aggregation map
Agg	Aggregation mesh
N_k	Number of local DOF
N	Number of total DOF
$\mathbf{Op}$	Operator matrix
k	Total degree of the polynomial space
α	Aggregation threshold
ϕ	Basis function
$\mathbb{P}_k$	Broken polynomial space
K	Numerical cell
K^{X}	Numerical cut-cell
X	Cut
Δt	Time step size
Γ_{D}	Edge imposed with Dirichlet boundary condition
Γ_{int}	Internal edge
Γ_{N}	Edge imposed with Neumann boundary condition
Γ	Edge
h	Numerical mesh size
$\mathfrak{K}_{\text{cc}}$	Set of cut-cells
$\mathfrak{K}_{\text{far}}$	Set of far-field cells
$\mathfrak{K}_{\text{near}}$	Set of cut-cell neighbours
$\mathfrak{K}_h^{\text{X}}$	Numerical cut-cell mesh
$\mathfrak{K}_h$	Numerical mesh
φ	Level-set function
Edg	Logical edge
$\mathbf{M}$	Mass matrix
$\mathbf{n}_\Gamma$	Edge normal field
$\mathbf{n}_{\mathfrak{I},\Gamma}$	Edge normal field including the interface normal vector
$\hat{F}$	Numerical flux
φ^{dg}	Level-set function represented by a DG field
φ^{c0}	Constrained level-set function represented by a DG field
η	Penalty parameter
$\mathbb{V}_{\mathbf{k}}^{\text{X}}$	Sum of broken polynomial spaces

ϑ	General test function
q	Test function for the continuity equation
r	Test function for the heat equation
$\mathbf{v}$	Test function for the momentum equation
Eo	Eötvös number
La	Laplace number
Pe	Peclet number
ω	Oscillation frequency
λ	Wavelength
k	Wavenumber

List of Abbreviations

AMR adaptive mesh refinement

BDF Backward Differentiation Formula

BoSSS Bounded Support Spectral Solver

CRC Collaborative Research Center

DG Discontinuous Galerkin

FEM Finite Element Method

FFT fast Fourier transformation

FVM Finite Volume Method

GNBC generalized Navier boundary condition

HMF Hierarchical Moment Fitting

IP interior penalty

IVP initial value problem

LDG local Discontinuous Galerkin

OP optimization problem

PDE partial differential equation

SIP symmetric interior penalty

XDG eXtended Discontinuous Galerkin

XFEM eXtended Finite Element Method

XNSE extended Navier-Stokes equations

1. Introduction

During the past decades numerical simulations emerged as a third approach between theory and experiments for the scientific study of complex physical phenomena. Especially in the context of computational fluid dynamics the importance is increasing, not at least due to the increase of computational power. Incompressible flow problems described by the Navier-Stokes equations only allow analytical solutions for reduced or simplified settings up to date. Considering in particular multi-phase flows the complexity is even increased due to:

- Discontinuous fluid properties across the interface, e.g. density and viscosity.
- Low-regularity solutions of flow properties, i.e. kinks in the velocity field and jumps in the pressure field.
- The presence of interfacial forces, e.g. force induced by surface tension.

Most numerical methods established in research and industry are based on the finite volume method (FVM). However, such low order methods suffer form their inherent property that for acquiring a certain accuracy the computational costs increase faster than the corresponding accuracy. This is even more severe when considering three-dimensional problems. A remedy for that are high-order methods such as the finite element method (FEM) and Discontinuous Galerkin (DG) method. However, FEM exhibits global mass matrices, which is unfavourable for local mesh refinement. DG methods allow a combination of both methods: a locally conservative method with faster convergence rates, compared to the computational costs, and local adaptivity even on unstructured meshes.
However, the low-regularity solutions of multi-phase flows destroy the high-order property of standard DG methods. Therefore, an extended variant, i.e. the extended Discontinuous Galerkin (XDG) method, is used which adapts the approximation space conform to the interface in order to regain the favourable property. Thus, a high-order numerical method is available that allows direct numerical simulations of multiphase flows up to the position of the interface.

Such properties are especially needed in context of simulations with dynamic contact lines, where the interface touches the boundary. The physics in the direct area around the contact line, down to nanometre scales, is highly complex and up to date not fully understood. Therefore, the collaborative research centre (CRC) 1194 at the TU Darmstadt was funded for the fundamental analysis of the interaction between transport and wetting processes. Besides the momentum transport additionally heat and mass transport are investigated. Parts of the presented work have been developed during the first funding period.

1.1. Goals of this Work and Preliminary Studies

Regarding the need of action described above, the overall goal of the presented work is the development of a high-order transient multiphase flow solver on the basis of the XDG method featuring energy stability. The solver should reliably predict various surface tension driven flow problems including dynamic contact lines. In accordance to the research program of the CRC 1194 the solver should allow the simulation of coupled heat transfer problems with additional evaporation. The developments should be implemented within the open-source framework BoSSS (Bounded Support Spectral Solver), see next Section 1.2.

Related theses to the BoSSS framework are presented in order to clearly demarcate the own work. The basis of the framework, an incompressible Navier-Stokes solver and the XDG method were developed by Kummer (2012). The extension to inviscid compressible flows by means of a immersed boundary method is covered in Müller (2014), a low-Mach solver in Klein (2015) and unsteady compressible flows in Krämer-Eis (2017). XDG-specialized level-set methods were developed in Utz (2018) and the adaption of the immersed boundary method to particulate flows was done in Krause (2019).

1.2. The Open-Source Framework BoSSS

The open-source framework BoSSS (Bounded Support Spectral Solver) is under active development at the chair of fluid dynamics (FDY) at TU Darmstadt. This framework was initiated in 2008 as a research code in order to provide a general foundation for the scientific development, application and evaluation of high-order discretization schemes based on the DG/XDG method. Furthermore, it aims to overcome the gap between prototype codes with limited performance and generality on the one hand, and highly-optimized single-purpose research codes on the other hand. Thus, BoSSS is capable of rapid prototyping new discretizations for various physical problems, it is suitable for high performance computing due to its MPI implementation and it provides a sophisticated workflow and data management including continuous integration for automatic testing.

Over the years the BoSSS framework has been serving as the basis of many different solvers and corresponding publications. It was applied to compressible flows with the immersed boundary method (Müller et al., 2017; Geisenhofer et al., 2019), to low-Mach flows (Klein et al., 2016), and incompressible single-phase flows using the SIMPLE method (Klein et al., 2013; Klein et al., 2015) as well as immersed boundaries for particulate flows (Krause and Kummer, 2017). Furthermore, it features solvers for multiphase flows (Kummer, 2016; Gründing et al., 2020) and viscoelastic flows (Kikker and Kummer, 2018).

Since 2017 the BoSSS framework is publicly available under the Apache License at `http://github.com/FDYdarmstadt/BoSSS`.

1.3. Outline of this Work

The introduction stated the motivation and resulting goals for the present work. The scope regarding the developments and implementations are set within the framework BoSSS.

The governing continuum mechanical equations for this work are given in Chapter 2. The transient incompressible Navier-Stokes equations and heat equation for a sharp interface formulation are considered. The extension to contact line problems is treated by the generalized Navier boundary condition. At last the two-phase formulation is extended for a mass flux across the interface.

Chapter 3 represents the core of this work, where main aspects of the developed multi-phase flow solver are described. The basics for the standard and extended DG method are given and the corresponding spatial and temporal discretization are stated. Specialized methods for the interface representation and evolution are explained. Numerical details on the surface tension force, energy estimates and the overall solver structure are presented.

In Chapter 4 the presented flow solver is validated by various test cases involving comparisons to analytical solutions and benchmarks. Furthermore, numerical investigations regarding the stability of the implemented discretization are given.

The validation for contact line problems is provided in Chapter 5, where simulations of a droplet on a wall and the rise of liquid within a capillary are presented.

The extension of the flow solver for handling additional mass flux across the interface in context of evaporation problems is described in Chapter 6. The proposed spatial discretization is stated and details on the mass flux computations are explained. A validation against analytical solutions is shown.

A conclusion of the present work and potential further developments of the solver are discussed in Chapter 7.

2. Governing Equations for Transient Multi-Phase Flows with Sharp Interfaces

In this chapter the continuum mechanical description of the considered problem is introduced. For this work transient two-phase flows with a moving interface are discussed. The governing equations are formulated by a sharp interface representation. Let $\Omega \subset \mathbb{R}^D$, $D \in \{2, 3\}$, be some domain, see Figure 2.1, that is given as the disjoint partitioning of the time-dependent fluid bulk phases $\mathfrak{A}(t)$ and $\mathfrak{B}(t)$ and the moving interface $\mathfrak{I}(t)$:

$$\Omega = \mathfrak{A}(t) \,\dot\cup\, \mathfrak{I}(t) \,\dot\cup\, \mathfrak{B}(t). \tag{2.1}$$

The interface $\mathfrak{I}(t) := \overline{\mathfrak{A}}(t) \cap \overline{\mathfrak{B}}(t)$ is assumed to be a $(D-1)$-dimensional manifold.

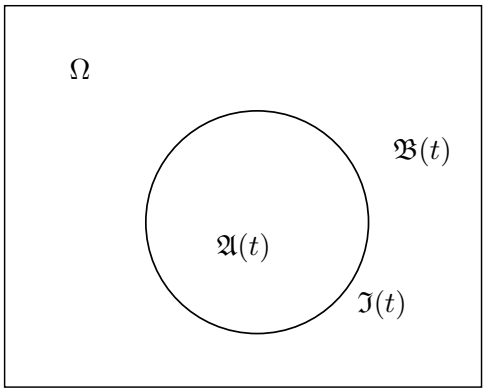


Figure 2.1.: Domain of interest Ω for a transient two-phase setting including the phases $\mathfrak{A}(t)$ and $\mathfrak{B}(t)$ with a moving interface $\mathfrak{I}(t)$.

In the first section, the balance equations for the single bulk phases, i.e. $\mathfrak{A}(t)$ and $\mathfrak{B}(t)$, are given. Then, the dividing interface $\mathfrak{I}(t)$ as a singular surface is introduced in Section 2.2, and the balance equations on the interface are presented in Section 2.3. Furthermore, the extension to dynamic contact line problems is discussed in Section 2.4 and the coupling of the heat equation with additional evaporation, i.e. a mass flux at the interface, in Section 2.5.

Main parts of this chapter are based on the works of Wang and Oberlack (2011) and Gross and Reusken (2011). For more details on the following outline the reader is referred to these works. Further information on the presented topics are found in Panton (2013), Slattery et al. (2007), and Kaviany (2001).

All equations in this chapter are given in the strong formulation, the weak formulations for the numerical representation are presented in the next chapter.

2.1. Balance Equations for the Bulk Phases

In this section the relevant balance equations for this work are introduced on the bulk domain $\Omega \setminus \mathfrak{I}$, i.e. $\mathfrak{A}$ and $\mathfrak{B}$. Here, the considered domains are time-independent and the boundary $\partial\Omega$ is given as a disjoint decomposition $\partial\Omega = \partial\Omega_{\mathrm{D}} \dot\cup \partial\Omega_{\mathrm{N}}$ with one part imposed by Dirichlet conditions $\partial\Omega_{\mathrm{D}}$ and a second part by Neumann conditions $\partial\Omega_{\mathrm{N}}$.

Furthermore, all occurring fields or physical properties, in general denoted by $\psi(\mathbf{x})$, are assumed to be piecewise defined in $\Omega \setminus \mathfrak{I}$, resp. $\mathfrak{A}$ and $\mathfrak{B}$. Thus it is defined that

$$\psi(\mathbf{x}) = \begin{cases} \psi_{\mathfrak{A}} \text{ for } \mathbf{x} \in \mathfrak{A}, \\ \psi_{\mathfrak{B}} \text{ for } \mathbf{x} \in \mathfrak{B}. \end{cases} \tag{2.2}$$

2.1.1. Incompressible Navier-Stokes Equations

The considered fluid dynamics in the bulk are described by the transient incompressible Navier-Stokes equations, including the continuity equation, which are given as follows

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) = \nabla \cdot \left(-p\mathbf{I} + \mu \left(\nabla \mathbf{u} + \nabla \mathbf{u}^{\mathrm{T}} \right) \right) + \rho \mathbf{f} \qquad \text{in } \Omega \setminus \mathfrak{I}, \tag{2.3a}$$

$$\nabla \cdot \mathbf{u} = 0 \qquad \text{in } \Omega \setminus \mathfrak{I}. \tag{2.3b}$$

Here, equation (2.3a) describes the momentum balance equations, where $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$ is the velocity vector and $p = p(\mathbf{x}, t)$ the pressure. The physical properties density and dynamic viscosity are denoted by ρ and μ, respectively, which are assumed to be constant for each phase. On the right-hand side $\mathbf{f} = \mathbf{f}(\mathbf{x}, t)$ describes a mass specific force density field, e.g. gravity. The considered fluid is a Newtonian viscous one, where the stress tensor $\mathbf{S}$ for incompressible fluids takes the form of $\mathbf{S} = -p\mathbf{I} + 2\mu\mathbf{D}(\mathbf{u})$ with the rate of deformation tensor $\mathbf{D}(\mathbf{u}) = \frac{1}{2}\left(\nabla \mathbf{u} + \nabla \mathbf{u}^{\mathrm{T}}\right)$. Equation (2.3b) describes the balance of mass for $\rho = \text{const}$, which is also called in the following the continuity equation.

At the boundary $\partial\Omega = \partial\Omega_{\mathrm{D}} \dot\cup \partial\Omega_{\mathrm{N}}$ the corresponding boundary conditions read

$$\mathbf{u} = \mathbf{u}_{\mathrm{D}} \qquad \text{on } \partial\Omega_{\mathrm{D}}, \tag{2.4a}$$

$$-p\mathbf{n}_{\partial\Omega} + \mu \left(\nabla \mathbf{u} + \nabla \mathbf{u}^{\mathrm{T}} \right) \mathbf{n}_{\partial\Omega} = -p_{\mathrm{ext}} \mathbf{n}_{\partial\Omega} \qquad \text{on } \partial\Omega_{\mathrm{N}} \tag{2.4b}$$

with given functions $\mathbf{u}_{\mathrm{D}}$ and p_{ext}. The normal $\mathbf{n}_{\partial\Omega}$ is outward pointing on the boundary $\partial\Omega$. Since we are considering a time-dependent problem, the initial value problem requires to set an initial condition for the velocity by

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}) \quad \text{ for } \mathbf{x} \in \Omega \setminus \mathfrak{I} \quad \text{ with } \nabla \cdot \mathbf{u}_0 = 0. \tag{2.5}$$

2.1.2. Total Energy Balance Equation

A direct result from the momentum equations (2.3a) is the balance equation for the kinetic energy e_{kin}, which we define for the incompressible case as the specific quantity $e_{\text{kin}} = \frac{1}{2}\|\mathbf{u}\|^2$. Scalar multiplication of the momentum equations (2.3a) with the velocity $\mathbf{u}$ and rewriting in divergence form results in

$$\rho \frac{\mathrm{D}e_{\text{kin}}}{\mathrm{D}t} = \nabla \cdot (\mathbf{S}\mathbf{u}) - \operatorname{tr}(\mathbf{S}\nabla\mathbf{u}) + \rho\vec{f} \cdot \mathbf{u}, \tag{2.6}$$

where $\frac{\mathrm{D}}{\mathrm{D}t}$ denotes the material derivative for a scalar field property ψ with $\frac{\mathrm{D}\psi}{\mathrm{D}t} = \frac{\partial\psi}{\partial t} + \mathbf{u} \cdot \nabla\psi$. The trace term on the right-hand side describes a dissipation term, which is found with opposite sign in the balance equation for the internal energy e_{int} with

$$\rho \frac{\mathrm{D}e_{\text{int}}}{\mathrm{D}t} = -\nabla \cdot \mathbf{q} + \operatorname{tr}(\mathbf{S}\nabla\mathbf{u}), \tag{2.7}$$

where $\mathbf{q}$ denotes the heat flux vector and the radiant flux is neglected. Thus both energy quantities e_{kin} and e_{int} are not conserved. However the total energy with $e_{\text{tot}} = e_{\text{int}} + e_{\text{kin}}$ is a conserved quantity with

$$\rho \frac{\mathrm{D}e_{\text{tot}}}{\mathrm{D}t} = -\nabla \cdot \mathbf{q} + \nabla \cdot (\mathbf{S}\mathbf{u}) + \rho\mathbf{f} \cdot \mathbf{u}, \tag{2.8}$$

which is obtained by adding both energy balance equations (2.6) and (2.7).

2.1.3. Heat Equation

The balance equation for the internal energy (2.7) written in terms of the temperature T is known as the heat equation. The heat equation for a Newtonian fluid is given by

$$\rho c \frac{\mathrm{D}T}{\mathrm{D}t} = k\Delta T + \mu\mathbf{D} : \mathbf{D}. \tag{2.9}$$

In this case, the heat flux is assumed to be isotropic and defined via Fourier's law as $\mathbf{q} = -k\nabla T$, where k denotes the thermal conductivity. For incompressible flows the specific heat capacity at constant pressure c_p is equal to the one at constant volume c_v, thus we denote the specific heat capacity by c. Again the physical properties are assumed to be constant in each domain. The last term on the right-hand side describes a dissipation term. The corresponding Dirichlet and Neumann boundary conditions at $\partial\Omega = \partial\Omega_{\text{D}} \dot{\cup} \partial\Omega_{\text{N}}$ are given by

$$T = T_D \qquad \text{on } \partial\Omega_{\text{D}}, \tag{2.10a}$$

$$-k\nabla T \cdot \mathbf{n}_{\partial\Omega} = \mathbf{q}_{\text{N}} \cdot \mathbf{n}_{\partial\Omega} \qquad \text{on } \partial\Omega_{\text{N}}. \tag{2.10b}$$

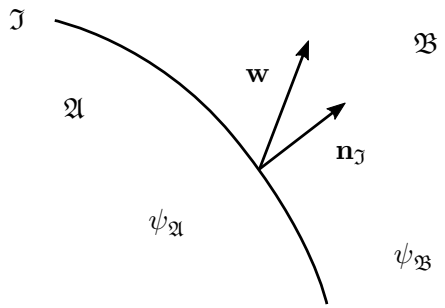


Figure 2.2.: Two-phase setting with the interface $\mathfrak{I}$ described as a singular surface.

2.2. The Sharp Interface Formulation

Before stating the balance equations defined on the interface, the two-phase setting for a sharp interface formulation is introduced. In this case the interface $\mathfrak{I} := \overline{\mathfrak{A}} \cap \overline{\mathfrak{B}}$ is described as a $(D-1)$-dimensional singular surface in $\Omega \subset \mathbb{R}^D$, for $D = 2$ see Figure 2.2.

The singular surface, i.e. the interface $\mathfrak{I}$, divides the whole domain Ω into two disjoint subdomains $\mathfrak{A}$ and $\mathfrak{B}$. Here, one can define an oriented normal field $\mathbf{n}_\mathfrak{I}$ on $\mathfrak{I}$, which is directed from $\mathfrak{A}$ to $\mathfrak{B}$. We assume that an arbitrary field property $\boldsymbol{\psi} \in \mathbb{R}^D$ is sufficiently continuous and differentiable in $\Omega \setminus \mathfrak{I}$, but it is allowed to be discontinuous across the interface. Thus a jump operator $[\![\cdot]\!](\mathbf{x})$ on $\mathbf{x} \in \mathfrak{I}$ can be defined as follows

$$[\![\boldsymbol{\psi}]\!](\mathbf{x}) := \boldsymbol{\psi}_{\mathfrak{B},\mathfrak{I}}(\mathbf{x}) - \boldsymbol{\psi}_{\mathfrak{A},\mathfrak{I}}(\mathbf{x}), \tag{2.11}$$

where the one-sided limits exist, i.e. ψ is sufficiently continuous and differentiable up to the interface, with

$$\boldsymbol{\psi}_{\mathfrak{A},\mathfrak{I}}(\mathbf{x}) := \lim_{\xi \searrow 0} \boldsymbol{\psi}(\mathbf{x} - \xi \mathbf{n}_\mathfrak{I}) \quad \forall \mathbf{x} \in \mathfrak{I} \tag{2.12a}$$

$$\boldsymbol{\psi}_{\mathfrak{B},\mathfrak{I}}(\mathbf{x}) := \lim_{\xi \searrow 0} \boldsymbol{\psi}(\mathbf{x} + \xi \mathbf{n}_\mathfrak{I}) \quad \forall \mathbf{x} \in \mathfrak{I} \tag{2.12b}$$

The singular surface in general moves with its own velocity $\mathbf{w}$, which is different from both material velocities $\mathbf{u}_{\mathfrak{A},\mathfrak{I}}$ and $\mathbf{u}_{\mathfrak{B},\mathfrak{I}}$ directly at the interface. Such a case occurs when considering evaporation at the interface, see Section 2.5.

There are two descriptions for the interface representation: First, an explicit formulation via a parameter set $Q \subset \mathbb{R}^{D-1}$ and a corresponding parametrization defined by the mapping function $F(\mathbf{q}, t)$ with $\mathbf{q} \in Q$ such that the interface $\mathfrak{I}$ is given by the set $F(\overline{Q}, t)$. Second, an implicit form given by a function φ with

$$\varphi(\mathbf{x}, t) = 0, \quad \mathbf{x} \in \mathfrak{I}. \tag{2.13}$$

In this case, an evolution equation for the interface $\mathfrak{I}$ is derived by differentiation w.r.t time resulting in

$$\frac{\partial \varphi}{\partial t} + \mathbf{w} \cdot \nabla \varphi = 0. \tag{2.14}$$

2.3. Balance Equations at the Interface

For the subsequent balance equations at the interface, it is assumed that the interface itself does not carry any additional mass. Furthermore, it is assumed that the interface is a material singular surface with $\mathbf{w} = \mathbf{u}_{\mathfrak{A},\mathfrak{I}} = \mathbf{u}_{\mathfrak{B},\mathfrak{I}}$. Note that the term "material" does not include that there is actually mass on the interface.

Two-phase Incompressible Navier-Stokes Equations The mass balance at the interface for no additional surface mass is given by

$$[\![\rho\left(\mathbf{u}-\mathbf{w}\right)\cdot\mathbf{n}_{\mathfrak{I}}]\!] = 0, \tag{2.15}$$

and assuming a material interface it is stated as

$$[\![\mathbf{u}]\!] = 0. \tag{2.16}$$

This condition furthermore states that there is no slip at the interface. The balance equation for momentum at a material interface is given by

$$\left[\!\left[-p\mathbf{n}_{\mathfrak{I}} + \mu\left(\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathrm{T}}\right)\mathbf{n}_{\mathfrak{I}}\right]\!\right] = \sigma\kappa\mathbf{n}_{\mathfrak{I}}, \tag{2.17}$$

where σ denotes the surface tension coefficient that is assumed to be constant. The quantity κ describes the mean curvature of the interface $\mathfrak{I}$ and is defined as

$$\kappa := \nabla_{\mathfrak{I}}\cdot\mathbf{n}_{\mathfrak{I}}. \tag{2.18}$$

The interface gradient operator $\nabla_{\mathfrak{I}}$ is defined as $\nabla_{\mathfrak{I}} := \mathbf{P}_{\mathfrak{I}}\nabla$, where $\mathbf{P}_{\mathfrak{I}}$ describes the projection on the interface $\mathfrak{I}$ with $\mathbf{P}_{\mathfrak{I}} := \mathbf{I} - \mathbf{n}_{\mathfrak{I}}\otimes\mathbf{n}_{\mathfrak{I}}$.

Two-Phase Energy Balance Equations The interface balance equation of the total energy at a material interface, i.e no surface mass and surface heat flux, reads

$$[\![\mathbf{q}\cdot\mathbf{n}_{\mathfrak{I}}]\!] - \mathbf{u}_{\mathfrak{I}}\cdot[\![\mathbf{S}\cdot\mathbf{n}_{\mathfrak{I}}]\!] = \sigma\nabla_{\mathfrak{I}}\cdot\mathbf{u}_{\mathfrak{I}} - \sigma\kappa\left(\mathbf{u}_{\mathfrak{I}}\cdot\mathbf{n}_{\mathfrak{I}}\right). \tag{2.19}$$

Note that $\nabla_{\mathfrak{I}}\cdot\mathbf{u}_{\mathfrak{I}} - \kappa\left(\mathbf{u}_{\mathfrak{I}}\cdot\mathbf{n}_{\mathfrak{I}}\right) = \nabla_{\mathfrak{I}}\cdot\mathbf{u}_{\mathfrak{I},\|}$, where $\mathbf{u}_{\mathfrak{I},\|}$ denotes the tangential component of the interface velocity.

The internal energy balance equation on the interface can be regained by multiplying the interface momentum balance equation (2.17) with the interface velocity $\mathbf{u}_{\mathfrak{I}}$, i.e. resulting in the kinetic energy balance on the interface, and by subtracting the result from the total energy balance (2.19). Thus, the interface balance equation reads

$$[\![\mathbf{q}\cdot\mathbf{n}_{\mathfrak{I}}]\!] = \sigma\nabla_{\mathfrak{I}}\cdot\mathbf{u}_{\mathfrak{I}}. \tag{2.20}$$

Two-Phase Heat Equation Considering the two-phase heat equation, we assume that the temperature at the interface is continuous, i.e. $[\![T]\!] = 0$. The corresponding interface balance equation without mass flux yields

$$[\![\mathbf{q}\cdot\mathbf{n}_{\mathfrak{I}}]\!] = [\![-k\nabla T\cdot\mathbf{n}_{\mathfrak{I}}]\!] = 0 \tag{2.21}$$

2.4. Two-Phase Flows with Dynamic Contact Lines

So far it was implicitly assumed that the interface $\mathfrak{I}$ is a closed surface with $\partial\mathfrak{I} = \emptyset$. However considering for example a droplet on the wall, see Figure 2.3, the interface forms a so-called three-phase contact line $L(t) = \partial\mathfrak{I}$ on the wall boundary $\partial\Omega_{\text{wall}}$. Here, both phases $\mathfrak{A}$ and $\mathfrak{B}$ have contact with the wall in one point for $\Omega \subset \mathbb{R}^2$, or along a curve if $\Omega \subset \mathbb{R}^3$.

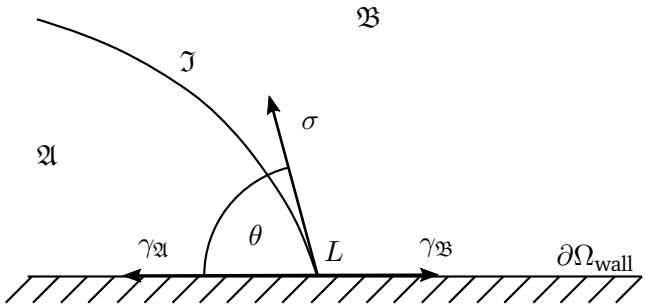


Figure 2.3.: Contact line setting on a wall $\partial\Omega_{\text{wall}}$, where the surface tension force σ of the interface $\mathfrak{I}$ and of both phases $\mathfrak{A}$ and $\mathfrak{B}$, i.e. $\gamma_{\mathfrak{A}}$ and $\gamma_{\mathfrak{B}}$, are acting directly at the contact line L. A contact angle θ is formed between the interface $\mathfrak{I}$ and the wall $\partial\Omega_{\text{wall}}$.

Between the wall and the interface the formation of an apparent contact angle θ in the denser phase $\mathfrak{A}$ is visible. Considering a static droplet, this contact angle takes a specific value θ_{stat} according to Young's equation (Young, 1805)

$$\sigma \cos\theta_{\text{stat}} = \gamma_{\mathfrak{B}} - \gamma_{\mathfrak{A}}. \tag{2.22}$$

This equation describes the equilibrium force balance tangential to the boundary $\partial\Omega_{\text{wall}}$ at the contact line L, where $\gamma_{\mathfrak{A}}$ and $\gamma_{\mathfrak{B}}$ denote the surface tension coefficients between the wall and the phases $\mathfrak{A}$ and $\mathfrak{B}$, respectively. Thus, for a given setting of σ, $\gamma_{\mathfrak{A}}$ and $\gamma_{\mathfrak{B}}$, the so-called static contact angle θ_{stat} is adjusted.

Such a static setting is well-posed with the standard no-slip boundary condition within a continuum mechanical modelling. However, considering a moving contact line, e.g. a droplet on a tilted wall, the standard approach is no longer well-posed. Huh and Scriven (1971) show that the no-slip boundary condition leads to a diverging stress field, which is non-integrable when approaching the moving contact line. This results further in a non-physical divergence in the energy dissipation rate. In order to regularize the singularity at the moving contact line the introduction of a Navier-slip boundary condition (Navier, 1823) may be used, e.g. see Huh and Scriven (1971) and Dussan V. (1976). However, this relaxation of the no-slip boundary condition still features a weak, i.e. integrable, singularity of the pressure field (Sprittles and Shikhmurzaev, 2011). Another feature of slip models is that they exhibit low-velocity regions near the contact line, see Shikhmurzaev (2006). However, such non-physical stagnation regions prevent the development of a rolling motion, which are first observed in Dussan V. and Davis (1974) and further investigated in Dussan V. (1976) and Chen et al. (1997a). An approach aiming to remove both artificial features is introduced with the interface formation model (IFM) by Shikhmurzaev (1993) and Shikhmurzaev (2007). This model extends the continuum

mechanical description via additional surface mass densities at the interface between all three phases. For each pair individual mass and momentum balance equations are incorporated and the surface tension also depends on the surface mass distribution. A reduced model that relaxes the impermeability condition and allows mass exchange between the interfaces and the bulk area is presented in Lukyanov and Pryer (2017). Nevertheless, the use of a Navier-slip boundary condition in context of macroscopic flow simulations is widespread, e.g. see Bao et al. (2012), Gao and Wang (2014), and Shen et al. (2015). In this work we also impose the Navier-slip boundary condition, more precisely the generalized Navier boundary condition, for flow problems with dynamic contact lines.

2.4.1. Generalized Navier Boundary Condition

The problem of the dynamic contact angle is handled by the use of a slip boundary condition, namely the generalized Navier boundary condition (GNBC). In this section the derivation of this boundary condition is outlined following the work of Reusken et al. (2017). The idea of this class of boundary conditions is based on the introduction of appropriate (virtual) dissipative effective forces at the slip wall $\partial\Omega_S$ and the contact line L. At $\partial\Omega_S$ one no longer assumes the no-slip condition, but the no-penetration condition with $\mathbf{u} \cdot \mathbf{n}_{\partial\Omega} = 0$ is still imposed. The slip at the wall is determined by a tangential force $\mathbf{f}_S$, which satisfies the following force balance condition implied by the momentum conservation

$$\mathbf{f}_S = \mathbf{P}_S(-p\mathbf{I} + \mu(\nabla\mathbf{u} + \nabla\mathbf{u}^T))\mathbf{n}_{\partial\Omega} = \mu\mathbf{P}_S(\nabla\mathbf{u} + \nabla\mathbf{u}^T)\mathbf{n}_{\partial\Omega} \quad \text{on } \partial\Omega_S, \tag{2.23}$$

where $\mathbf{P}_S := \mathbf{I} - \mathbf{n}_{\partial\Omega} \otimes \mathbf{n}_{\partial\Omega}$ denotes the orthogonal projection onto the slip wall $\partial\Omega_S$. For the modelling of $\mathbf{f}_S$ the reader is referred to Ren and E (2007), in which molecular dynamics (MD) is used to derive effective boundary conditions for the continuum mechanical formulations at the contact line. Their results give the following description for a dissipative friction force

$$\mathbf{f}_S = \mathbf{f}_{S,\text{diss}}(\mathbf{u}) = -\beta_S\mathbf{P}_S\mathbf{u} \tag{2.24}$$

with the phase coefficient of friction $\beta_S \geq 0$. Equations (2.23) and (2.24) in combination with the no-penetration condition lead to the standard Navier-slip boundary condition:

$$\mathbf{u} \cdot \mathbf{n}_S = 0 \qquad \text{on } \partial\Omega_S, \tag{2.25a}$$

$$\mu\mathbf{P}_S(\nabla\mathbf{u} + \nabla\mathbf{u}^T)\mathbf{n}_S = -\beta_S\mathbf{P}_S\mathbf{u} \qquad \text{on } \partial\Omega_S. \tag{2.25b}$$

Note that the friction coefficient introduces a slip length l_s with $\beta_S = \frac{\mu}{l_s}$. For $\beta_S = 0$ the Navier-slip boundary condition reduces to a free-slip boundary condition.

For a moving contact line $L(t)$ an additional dissipative force $\mathbf{f}_{L,\text{diss}} = -\beta_L U_L \mathbf{n}_L$ with $\beta_L \geq 0$ is introduced, where $\mathbf{n}_L$ is the normal to L and tangential to the wall $\partial\Omega_S$. Further, $U_L = \mathbf{u} \cdot \mathbf{n}_L$ denotes the contact line velocity. In combination with Young's equation (2.22) the resulting effective force at the contact line is given by

$$\mathbf{f}_L = -\beta_L U_L \mathbf{n}_L + \sigma\cos\theta_{\text{stat}}. \tag{2.26}$$

Again, the momentum conservation implies a force balance condition, which is now located at L with

$$\mathbf{P}_S\mathbf{S}_\mathfrak{I}\boldsymbol{\tau}_L = \mathbf{f}_L, \tag{2.27}$$

where $\boldsymbol{\tau}_L = \frac{\mathbf{P}_\mathfrak{I}\mathbf{n}_{\partial\Omega}}{\|\mathbf{P}_\mathfrak{I}\mathbf{n}_{\partial\Omega}\|}$ is the normal to L and tangential to $\mathfrak{I}$. The interface stress tensor is given by $\mathbf{S}_\mathfrak{I} = \sigma\mathbf{P}_\mathfrak{I}$. Here, the left-hand side of equation (2.27) can be rewritten to $\mathbf{P}_\mathrm{S}\mathbf{S}_\mathfrak{I}\boldsymbol{\tau}_L = \sigma\mathbf{P}_\mathrm{S}\mathbf{P}_\mathfrak{I}\boldsymbol{\tau}_L = \sigma\mathbf{P}_\mathrm{S}\boldsymbol{\tau}_L = \sigma\cos(\theta)\mathbf{n}_L$ and thus the force balance (2.27) with equation (2.26) reads

$$\sigma(\cos\theta_\mathrm{stat} - \cos\theta) = \beta_L U_L \quad \text{on } L, \tag{2.28}$$

where θ describes the current contact angle. For $\beta_L = 0$ the force balance (2.28) in combination with the Navier-slip boundary condition (2.25) results in the generalized Navier boundary condition, which describes a quasi-static contact angle model (Reusken et al., 2017). Such models implicitly impose the static contact at all times and any deviation from this should come from discretization errors (Buscaglia and Ausas, 2011). Values of $\beta_L > 0$ allow a relation between the current contact angle θ and the contact line velocity U_L, which leads to a dynamic contact angle model.

2.5. Two-Phase Flows with Mass Flux at the Interface

Considering the mass balance at the interface with no additional surface mass (2.15), this equation can be split into the following form

$$\rho_\mathfrak{A}\left((\mathbf{u}_{\mathfrak{A},\mathfrak{I}} - \mathbf{w})\cdot\mathbf{n}_\mathfrak{I}\right) = \rho_\mathfrak{B}\left((\mathbf{u}_{\mathfrak{B},\mathfrak{I}} - \mathbf{w})\cdot\mathbf{n}_\mathfrak{I}\right) = \dot{m}, \tag{2.29}$$

where $\dot{m}$ denotes the mass transfer rate across the interface. It is such defined, that an inflow of mass for domain $\mathfrak{A}$ is negative and positive for an outflow. Eliminating the surface velocity $\mathbf{w}$, the mass balance in terms of the mass transfer rate is given by

$$[\![\mathbf{u}]\!]\cdot\mathbf{n}_\mathfrak{I} = -\dot{m}\left[\!\left[\rho^{-1}\right]\!\right]. \tag{2.30}$$

Assuming a non-material interface for the momentum balance, equation (2.17) extends to

$$\left[\!\left[\rho\mathbf{u}\left(\mathbf{w}-\mathbf{u}\right)\cdot\mathbf{n}_\mathfrak{I} - p\mathbf{n}_\mathfrak{I} + \mu\left(\nabla\mathbf{u} + \nabla\mathbf{u}^\mathrm{T}\right)\mathbf{n}_\mathfrak{I}\right]\!\right] = \sigma\kappa\mathbf{n}_\mathfrak{I}, \tag{2.31}$$

which by applying the definitions of the mass transfer rate in equations (2.29) is stated as

$$-\dot{m}\,[\![\mathbf{u}]\!] + \left[\!\left[-p\mathbf{n}_\mathfrak{I} + \mu\left(\nabla\mathbf{u} + \nabla\mathbf{u}^\mathrm{T}\right)\mathbf{n}_\mathfrak{I}\right]\!\right] = \sigma\kappa\mathbf{n}_\mathfrak{I}. \tag{2.32}$$

Using the mass balance equation (2.30), the momentum balance at the interface results in

$$\dot{m}^2\left[\!\left[\rho^{-1}\right]\!\right]\mathbf{n}_\mathfrak{I} + \left[\!\left[-p\mathbf{n}_\mathfrak{I} + \mu\left(\nabla\mathbf{u} + \nabla\mathbf{u}^\mathrm{T}\right)\mathbf{n}_\mathfrak{I}\right]\!\right] = \sigma\kappa\mathbf{n}_\mathfrak{I}. \tag{2.33}$$

The extension of the energy balance equations (2.19) and (2.20) for a non-material interface is not discussed in this work. However, in this work the simulation of two-phase flow problem with evaporation is considered, so the two-phase heat equation with additional mass flux is presented.

Two-phase Heat Equation with Mass Flux at the Interface As described above, we assume that the temperature is continuous everywhere at the interface. In context of evaporation problems we further assume that the temperature for both phases at the interface is equal to the saturation temperature T_{sat} corresponding to the system bulk pressure p_0 given by

$$T = T_{\text{sat}}(p_0), \quad \text{on } \mathfrak{I}. \tag{2.34}$$

The interface balance equation reads

$$[\![\mathbf{q} \cdot \mathbf{n}_{\mathfrak{I}}]\!] = [\![-k\nabla T \cdot \mathbf{n}_{\mathfrak{I}}]\!] = \dot{m}\left(h_{\text{vap}} + \frac{\dot{m}^2}{2}\left[\!\left[\rho^{-2}\right]\!\right]\right), \tag{2.35}$$

where for moderate transfer rates the convective contribution $\frac{\dot{m}^2}{2}\left[\!\left[\rho^{-2}\right]\!\right]$ on the right-hand side is small compared to the latent heat of vaporization h_{vap}. Note that the balance equation (2.35) reduces to (2.21) for $\dot{m} = 0$.

3. Discontinuous Galerkin Discretization for Transient Multi-Phase Flows with Sharp Interfaces

In this chapter the numerical basics of the developed transient multi-phase flow solver based on the Extended Discontinuous Galerkin (XDG) method is explained. The spatial discretization of the governing equations stated in Chapter 2 are given in Section 3.4. The corresponding temporal discretization for transient flows with a moving interface is presented in Section 3.5. The level-set representation and in this work used specialized methods for the interface evolution in context of XDG are found in Section 3.6. Details on the discretization of the surface tension force are discussed in Section 3.7 and energy estimates for the presented discretization are given in Section 3.8. This chapter is concluded by an overview on the overall solver structure, see Section 3.9.

3.1. State of the Art

Discontinuous Galerkin (DG) methods have gained in recent years quite some interest in context of computational fluid dynamics. Besides the well known finite volume method (FVM) and finite element method (FEM), DG methods allow the combination of favourable properties of both FVM and FEM. Like FVM, DG discretizations are locally conservative and exhibit a flux formulation that directly reflects the underlying physical problem dynamics. Furthermore, DG enables a high-order method even on unstructured meshes by adapting the local ansatz functions, whereas standard FVM needs to construct larger stencils of neighbouring cells. This locality property also marks an advantage against FEM that defines global mass matrices, which are costly to invert. DG discretizations result in local mass matrices and sparse operator matrices. However, a major drawback of DG methods is the notably increase of degrees of freedom, when comparing to FVM on the same grid and interpolation order.

The DG method was first introduced by Reed and Hill (1973) to solve the steady-state neutron transport equation and analyzed in Lesaint and Raviart (1974). The latter showed an optimal convergence rate up to (h^{k+1}) for rectangles of mesh size h and with an polynomial approximation order of k. An extension to hyperbolic conservation laws are presented in Cockburn and Shu (1989) and for convection-diffusion systems in Cockburn and Shu (1998). For the latter the second-order operator is reformulated as a system of first-order equations leading to local discontinuous Galerkin methods (LDG). An alternative approach without reformulation provide (symmetric) interior penalty (SIP/IP) methods, which are firstly introduced in context

of elliptic problems by Arnold (1982) and Douglas and Dupont (1976). Further developments in context of DG regarding the stability of the IP methods are given in Brezzi et al. (2006) and Shahbazi (2005). Both approaches allowed the extension of the DG method to new applications such as incompressible fluid flows, see Cockburn et al. (2004), Cockburn et al. (2005), Liu and Shu (2000), Girault et al. (2004), and Shahbazi et al. (2007). The latter used a local Lax-Friedrichs flux for the convective terms of the Navier-Stokes equations in combination with the SIP method.

Extended Methods for Finite Elements Considering the discretization of the sharp interface two-phase Navier-Stokes equations (2.3), (2.16) and (2.17) with the DG method and allowing the occurence of discontinuous parameters at the interface, one need to adapt the approximation space conformal with the position of the interface. However, in context of flow problems one wants to avoid a remeshing every time step in order to adapt the solution space. A first method that enriches the approximation space such that the solution may exhibit discontinuities inside a finite element is presented in Moës et al. (1999). This extended FEM (XFEM) was introduced for the simulation of crack growth in solid mechanics. The first application to incompressible two-phase flows is found Groß and Reusken (2007), where the pressure is discretized with XFEM and exhibits a jump due to surface tension. A discretization of both pressure and velocity with XFEM is provided by Fries (2009), which additionally allows the representation of the kink in the velocity field. However, the standard XFEM may lead to ill-conditioned system matrices due to enrichments with very small support (Reusken, 2008). The stable XFEM is proposed by Babuška and Banerjee (2012) and applied to two-phase flows in Sauerland and Fries (2013). Another extended approach in context of FEM is based on Nitsche's method (Nitsche, 1971) and was first derived in Hansbo and Hansbo (2002) for elliptic interface problems. In this case the continuity of the velocity field is enforced weakly at the interface. Additional stabilization for higher-order elements is provided by Burman (2010) with the introduction of face-oriented ghost-penalty terms. This stabilization is used for the steady Stokes problem in Hansbo et al. (2014) and unsteady two-phase Navier-Stokes equations in Schott et al. (2015). A space-time approach for a class of two-phase mass transport problems is presented in Lehrenfeld (2015), where a DG technique in time is combined with an XFEM. The jump condition is enforced weakly using the Nitsche method (Lehrenfeld and Reusken, 2012; Lehrenfeld and Reusken, 2013).

The first extended method for DG is presented in Bastian and Engwer (2009) for the discretization of elliptic scalar problems. In Heimann et al. (2013) this approach is applied to incompressible Navier-Stokes two-phase flows. The unfitted DG (UDG) method is based on the nonsymmetric interior penalty method, where the cut out meshes for both subdomains is based on a piece-wise linear approximation of the interface. The XDG method proposed by Kummer (2016) for steady two-phase flows uses a high-order approximation of the interface in combination with an quadrature technique for implicitly defined domains. The discretization is based on the SIP method and the stabilization against small cut-cells is ensured by cell agglomeration.

Another approach by Saye (2017) is the use of implicitly defined meshes with curved elements that are interface-conforming. The implicit mesh DG method provides high-order accuracy for the interface jump conditions in combination with the interfacial gauge method (Saye, 2016),

where the numerical coupling between the fluid velocity, pressure, and the interface position is reduced.

Considering the extension to flow problems including moving contact lines Reusken et al. (2017) presents a variational formulation of the GNBC for the stabilized XFEM. The interface is tracked implicitly by the level-set method which does not need any special boundary conditions. An extension of the UDG method is given in Heimann (2013) in the context of a conservative level-set method. In this case the boundary condition at the contact line need to be adapted for the recompression of the phase-field. Further works in the context of a finite element discretization with moving contact lines are found in Gerbeau and Lelièvre (2009), Manservisi and Scardovelli (2009), Ganesan and Tobiska (2009), and Ganesan (2013).

3.2. The Discontinuous Galerkin Method

This section provides an introduction to the non-extended Discontinuous Galerkin (DG) method. Basic definitions on standard notations are briefly summarized, the DG space and operators are given and the methods formulation for a general conservation equation is discussed. For a thorough introduction the reader is referred to the textbooks of Li (2006), Hesthaven and Warburton (2008), and Di Pietro and Ern (2012).

3.2.1. Basic Definitions

The following definitions are derived from Kummer (2016) and Kummer et al. (2020). Here and for the remainder of this work, it is assumed that the domain of interest Ω is a polygonal and simply connected domain. Thus, the computational domain $\Omega_h = \Omega$ and the subsequent definitions hold.

Definition 1 (Basic notations). For a polygonal and simply connected computational domain $\Omega \in \mathbb{R}^D$ ($D \in \{2, 3\}$) we define:

- the numerical (background) mesh $\mathfrak{K}_h = \{K_1, ..., K_J\}$ that covers the whole domain $\overline{\Omega} = \cup_j \overline{K}_j$ with non-overlapping cells ($\int_{K_j \cap K_l} 1 \, \mathrm{d}V = 0, l \neq j$), where h denotes the maximum diameter of all cells K_j.
- the set of all edges in the mesh is given by $\Gamma := \cup_j \partial K_j$. This set can be subdivided into $\Gamma = \Gamma_{\text{int}} \cup \Gamma_{\text{D}} \cup \Gamma_{\text{N}}$, where $\Gamma_{\text{int}} = \Gamma \setminus \partial\Omega$ denotes the set of all internal edges, Γ_{D} and Γ_{N} the set of edges imposed with Dirichlet and Neumann conditions, respectively.
- a normal field $\mathbf{n}_\Gamma$ on Γ, where it represents an outer normal on $\partial\Omega$, i.e. $\mathbf{n}_\Gamma = \mathbf{n}_{\partial\Omega}$.
- the broken gradient $\nabla_h \boldsymbol{\psi}$ defines for $\boldsymbol{\psi} \in C^1(\Omega \setminus \Gamma)$ the gradient on the domain $\Omega \setminus \Gamma$. According to that, the broken divergence $\nabla_h \cdot \boldsymbol{\psi}$ is defined.

For the approximation space of the ansatz and test functions, see next Section 3.2.2, we define the 'standard' DG space as a broken polynomial space.

Definition 2 (DG space). The broken polynomial space of total degree k is defined as:

$$\mathbb{P}_k(\mathfrak{K}_h) := \{\phi \in L^2(\Omega); \forall K \in \mathfrak{K}_h : \phi|_K \text{ is polynomial and } \deg(\phi|_K) \leq k\} \tag{3.1}$$

In general, DG methods are written in terms of jump and average operators defined at the edges Γ.

Definition 3 (inner and outer value, jump and average operator). For a field $\psi \in \mathcal{C}^0(\Omega \setminus \Gamma)$ we define the inner and outer values, ψ^{in} and ψ^{out}, at the edges Γ as:

$$\psi^{\text{in}}(\mathbf{x}) := \lim_{\xi \searrow 0} \psi(\mathbf{x} - \xi \mathbf{n}_\Gamma) \quad \text{for } \mathbf{x} \in \Gamma, \tag{3.2a}$$

$$\psi^{\text{out}}(\mathbf{x}) := \lim_{\xi \searrow 0} \psi(\mathbf{x} + \xi \mathbf{n}_\Gamma) \quad \text{for } \mathbf{x} \in \Gamma_{\text{int}}. \tag{3.2b}$$

Thus, the jump and average operator are defined as:

$$[\![\psi]\!] := \begin{cases} \psi^{\text{in}} - \psi^{\text{out}} & \text{on } \Gamma_{\text{int}} \\ \psi^{\text{in}} & \text{on } \partial\Omega \end{cases}, \tag{3.3}$$

$$\{\{u\}\} := \begin{cases} \frac{1}{2}(\psi^{\text{in}} + \psi^{\text{out}}) & \text{on } \Gamma_{\text{int}} \\ \psi^{\text{in}} & \text{on } \partial\Omega \end{cases}. \tag{3.4}$$

3.2.2. Discretization for a DG Method

As an introductory example, following Hesthaven and Warburton (2008), we are considering the discretization of a general conservation law with a non-linear flux function $\mathbf{f}(\psi)$ for a scalar quantity $\psi = \psi(\mathbf{x}, t)$ in Ω and suitable Dirichlet boundary condition on $\partial\Omega = \partial\Omega_{\text{D}}$ and a compatible initial condition ψ_0. The problem statement reads

$$\frac{\partial \psi}{\partial t} + \nabla \cdot \mathbf{f}(\psi) = 0, \qquad \mathbf{x} \in \Omega, \tag{3.5a}$$

$$\psi = \psi_{\text{D}}, \qquad \mathbf{x} \in \partial\Omega_{\text{D}}, \tag{3.5b}$$

$$\psi(\mathbf{x}, 0) = \psi_0(\mathbf{x}), \qquad \mathbf{x} \in \Omega. \tag{3.5c}$$

The goal is to find an approximate solution $\psi_h = \psi_h(\mathbf{x}, t)$ to ψ that fulfils the problem (3.5). Therefore, the problem domain Ω is discretized by a numerical mesh $\mathfrak{K}_h$ and for each numerical cell K_j we introduce the approximation by a local polynomial basis $\boldsymbol{\phi}_j = (\phi_{j,l})_{l=1,\ldots,N_k} \in \mathbb{P}_k(\{K_j\})$ with a cell-local support $\text{supp}(\boldsymbol{\phi}_j) = \overline{K}_j$ as

$$\psi_j(\mathbf{x}, t) = \sum_{l=1}^{N_k} \tilde{\psi}_{j,l}(t)\phi_{j,l}(\mathbf{x}) = \tilde{\boldsymbol{\psi}}_j(t) \cdot \boldsymbol{\phi}_j(\mathbf{x}), \quad \mathbf{x} \in K_j, \tag{3.6}$$

where the coefficients $\tilde{\boldsymbol{\psi}}_j = (\tilde{\psi}_{j,l})_{l=1,\ldots,N_k}$ denote the unknowns or degrees of freedom (DOF) of the local solution in cell K_j. In this work a modal polynomial basis is used, which fulfils the orthogonality condition

$$\int_{K_j} \phi_{j,m}\phi_{j,n} \, \mathrm{d}V = \delta_{mn} \tag{3.7}$$

with the Kronecker delta δ_{mn}. Inserting the local approximation (3.6) into the conservation law (3.5a) results in a cell-wise residual

$$R_j(\mathbf{x},t) = \frac{\partial \psi_j}{\partial t} + \nabla \cdot \mathbf{f}(\psi_j), \quad \mathbf{x} \in K_j. \tag{3.8}$$

For a Galerkin method, the residual (3.8) is minimized with respect to the same space as the ansatz functions, i.e. $\mathbb{P}_k(\{K_j\})$. Thus, we demand for the test functions $\vartheta_{j,l} = \phi_{j,l}$ in each cell $K_j \in \mathfrak{K}_h$ that

$$\int_{K_j} R_j \phi_{j,l} \, \mathrm{d}V = \int_{K_j} \frac{\partial \psi_j}{\partial t} \phi_{j,l} + \nabla \cdot \mathbf{f}(\psi_j) \phi_{j,l} \, \mathrm{d}V \overset{!}{=} 0, \quad \forall \boldsymbol{\phi}_{j,l}. \tag{3.9}$$

So, for each cell we end up with a linear system of N_k equations. However so far, no approximate global solution $\psi_h \in \Omega$ can be regained from the minimization in (3.9). The global solution is assumed to be a piecewise polynomial approximation

$$\psi(\mathbf{x},t) \approx \psi_h(\mathbf{x},t) = \bigoplus_{j=1}^{J} \psi_j(\mathbf{x},t) = \sum_{j=1}^{J} \sum_{l=1}^{N_k} \tilde{\psi}_{j,l}(t) \phi_{j,l}(\mathbf{x}) \in \mathbb{P}_k(\mathfrak{K}_h) \tag{3.10}$$

defined as the direct sum of the J local solutions ψ_j in (3.6). Here, $\tilde{\psi}_{j,l}$, with $j = 1, ..., J$ and $l = 1, ..., N_k$, denote the total DOF, with $N = J \cdot N_k$, of the global approximate solution ψ_h. In order to formulate a global DG method, the spatial term on the right-hand side of equation (3.9) is rewritten in terms of boundary edge integrals $\forall K_j \in \mathfrak{K}_h$ using partial integration

$$\int_{K_j} \frac{\partial \psi_j}{\partial t} \phi_{j,l} \, \mathrm{d}V - \int_{K_j} \mathbf{f}(\psi_j) \cdot \nabla_h \phi_{j,l} \, \mathrm{d}V + \oint_{\partial K_j} \left(\mathbf{f}(\psi_j) \cdot \mathbf{n}_j \right) \phi_{j,l} \, \mathrm{d}S = 0, \quad \forall \phi_{j,l}, \tag{3.11}$$

where $\mathbf{n}_j$ represents the local outward pointing normal for cell K_j. Note that on the internal edges Γ_{int} the value of $\mathbf{f}(\psi_j)$ is multiply defined. Therefore, a numerical flux $\hat{F}$ is introduced

$$\hat{F}(\psi_j^{\text{in}}, \psi_j^{\text{out}}, \mathbf{n}_\Gamma) \approx \mathbf{f}(\psi_j) \cdot \mathbf{n}_j, \tag{3.12}$$

that uniquely defines the resulting value of both neighbouring values, i.e. ψ_j^{in} and ψ_j^{out} at internal edges Γ_{int}. Summing over all cells K_j, the global minimization problem for $\psi_h(\mathbf{x},t), \mathbf{x} \in \Omega$ reads: Find $\psi_h \in \mathbb{P}_k(\mathfrak{K}_h)$, such that $\forall \vartheta = \phi \in \mathbb{P}_k(\mathfrak{K}_h)$

$$\int_\Omega \frac{\partial \psi_h}{\partial t} \phi \, \mathrm{d}V - \int_\Omega \mathbf{f}(\psi_h) \cdot \nabla_h \phi \, \mathrm{d}V + \oint_\Gamma \hat{F}(\psi_h^{\text{in}}, \psi_h^{\text{out}}, \mathbf{n}_\Gamma) \, [\![\phi]\!] \, \mathrm{d}S = 0, \tag{3.13}$$

where at Γ_{D} the outer value $\psi_h^{\text{out}} = \psi_{\text{D}}$ is given by the Dirichlet boundary condition (3.5b). In order to fully discretize the initial boundary value problem (3.5), one further needs to discretize the temporal term. This issue is skipped at this point and is discussed in Section 3.5. Thus, the current form of (3.13) is referred to as the semi-discrete weak formulation of (3.5).

Considering the spatial discretization, the numerical flux $\hat{F}$ needs to satisfy certain mathematical and physical properties to ensure stability and convergence of the DG method. In this work the stability is defined in the continuous setting via the energy estimate

$$\|\psi(\mathbf{x},t)\|_\Omega^2 \leq \|\psi(\mathbf{x},0)\|_\Omega^2, \quad \forall t \geq 0, \tag{3.14}$$

where homogenous Dirichlet conditions are assumed. Thus, stability is given, if the energy $\|\psi(\mathbf{x},t)\|_\Omega^2$ only decreases in the absence of an inflow. Two properties need to be fulfilled in order to proof that the discrete problem (3.13) satisfies the discrete equivalent of the energy estimate in (3.14). The numerical flux $\hat{F}$ is required to be a function that is Lipschitz continuous and monotonic, see Di Pietro and Ern (2012) for the proof. Furthermore, it is obvious that the DG method needs to regain a unique approximate solution to the underlying problem. Thus, $\hat{F}$ satisfies the following consistency property

$$\hat{F}(a,a,\mathbf{n}) = \mathbf{f}(a)\cdot\mathbf{n}, \quad \forall a \in \mathbb{R}. \tag{3.15}$$

A direct consequence of (3.15) is that the weak formulation (3.13) is directly fulfilled for $\psi_h = \psi$. Since considering a general conservation law in its conservative form, one further requires that $\hat{F}$ regains the global conservation property, i.e. the total amount of ψ only changes due to fluxes across the domain boundary $\partial\Omega$, by satisfying

$$\hat{F}(a,b,\mathbf{n}) = -\hat{F}(a,b,-\mathbf{n}), \quad \forall a,b \in \mathbb{R}. \tag{3.16}$$

The specific form of a suitable numerical flux $\hat{F}$ is presented in Section 3.4, where the spatial discretization of the two-phase flow problem is discussed.

Note that the system (3.13) can be written in a shorten matrix formulation as

$$\mathbf{M}\frac{\mathrm{d}\tilde{\boldsymbol{\psi}}}{\mathrm{d}t} + \mathbf{Op}(\tilde{\boldsymbol{\psi}}) = \mathbf{b}, \tag{3.17}$$

where the sought-after coefficients are given as the solution vector $\tilde{\boldsymbol{\psi}} = \{\tilde{\psi}_{1,1}, \tilde{\psi}_{1,2}, ..., \tilde{\psi}_{j,n}, ..., \tilde{\psi}_{J,N_k}\} \in \mathbb{R}^N$. The mass matrix $\mathbf{M} \in \mathbb{R}^{N\times N}$ has a block-diagonal structure with

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{M}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{M}_J \end{bmatrix}, \tag{3.18}$$

where the cell-local mass matrix $\mathbf{M}_j$ is defined by

$$(\mathbf{M}_j)_{m,n} = \int_{K_j} \phi_{j,m}\phi_{j,n}\,\mathrm{d}V = \int_{K_j} \boldsymbol{\phi}_j \otimes \boldsymbol{\phi}_j\,\mathrm{d}V. \tag{3.19}$$

The cell-local Operator matrix $\mathbf{Op}_j$ is given by

$$(\mathbf{Op}_j)_{m,n} = -\int_{K_j} \mathbf{f}(\tilde{\psi}_{j,n}\phi_{j,n})\cdot\nabla_h\phi_{j,m}\,\mathrm{d}V + \oint_{\partial K_j} \hat{F}(\tilde{\psi}_{j,n}, \tilde{\psi}_{j^*,n}, \mathbf{n}_\mathfrak{I})\phi_{j,m}\,\mathrm{d}S, \tag{3.20}$$

where j^* denotes the index of a neighbouring cell to K_j. Like the mass matrix, the operator matrix exhibits a block-diagonal structure, but including secondary diagonals due to the coupling with neighbouring cells over the numerical fluxes $\hat{F}$. The right-hand side $\mathbf{b}$ incorporates the given Dirichlet boundary condition value ψ_D.

3.3. The Extended Discontinuous Galerkin Method

In this section the extended DG method is introduced in order to allow the spatial discretization on an arbitrarily shaped domain, e.g. in context of two-phase flows. Main parts of this section are based on Kummer (2016) and Kummer et al. (2020). So far the computational domain Ω was assumed to be of polygonal shape covered by non-overlapping cells $\mathfrak{K}_h$. Now, in context of two-phase flows, we allow the existence of a dividing interface $\mathfrak{I}$ with $\Omega = \mathfrak{A}(t) \dot{\cup} \mathfrak{I}(t) \dot{\cup} \mathfrak{B}(t)$, see Equation (2.1). Before extending the basic definitions, see next Section 3.3.1, we first have to describe the interface. Therefore, we introduce a sufficiently smooth level-set function $\varphi(\mathbf{x}, t)$ that is almost everywhere $C^1(\Omega)$-continuous, but at least $\varphi(\mathbf{x}, t) \in C^0(\Omega)$. The partitioning of Ω is now implicitly defined such that

$$\mathfrak{A}(t) = \{\mathbf{x} \in \Omega : \varphi(\mathbf{x}, t) < 0\}, \tag{3.21a}$$

$$\mathfrak{B}(t) = \{\mathbf{x} \in \Omega : \varphi(\mathbf{x}, t) > 0\}, \tag{3.21b}$$

$$\mathfrak{I}(t) = \{\mathbf{x} \in \Omega : \varphi(\mathbf{x}, t) = 0\}. \tag{3.21c}$$

This level-set representation directly allows the formulation of the interface normal $\mathbf{n}_\mathfrak{I}$ via the gradient of φ by

$$\mathbf{n}_\mathfrak{I} = \frac{\nabla \varphi}{\|\nabla \varphi\|}. \tag{3.22}$$

Recall that the interface normal $\mathbf{n}_\mathfrak{I}$ is pointing from $\mathfrak{A}$ to $\mathfrak{B}$. Furthermore, the mean curvature can be regained by Bonnet's formula (2.18) with

$$\kappa = \nabla \cdot \left(\frac{\nabla \varphi}{\|\nabla \varphi\|} \right). \tag{3.23}$$

In the following subsections the extended DG space is introduced (Section 3.3.1) and numerical issues such as the integration on implicitly defined domains (Section 3.3.2) and stability against small cut-cells (Section 3.3.3) are discussed.

3.3.1. The Extended Discontinuous Galerkin Space

By introducing the interface $\mathfrak{I}$ into the computational domain, one can define cells that are associated solely to a certain phase $\mathfrak{s}(t) \in \{\mathfrak{A}(t), \mathfrak{B}(t)\}$. However, in cells where the interface is located both phases are present. Thus, we define cut-cells and a corresponding cut-cell mesh.

Definition 4 (cut-cells and cut-cell mesh). Time-dependent cut-cells are defined as

$$K_{j,\mathfrak{s}}(t) := K_j \cap \mathfrak{s}(t). \tag{3.24}$$

The set of all cut-cells $K_{j,\mathfrak{s}}(t)$ forms the time-dependent cut-cell mesh $\mathfrak{K}_h^{\mathrm{X}}(t) = \{K_{1,\mathfrak{A}}(t), K_{1,\mathfrak{B}}(t), \ldots, K_{J,\mathfrak{A}}(t), K_{J,\mathfrak{B}}(t)\}$.

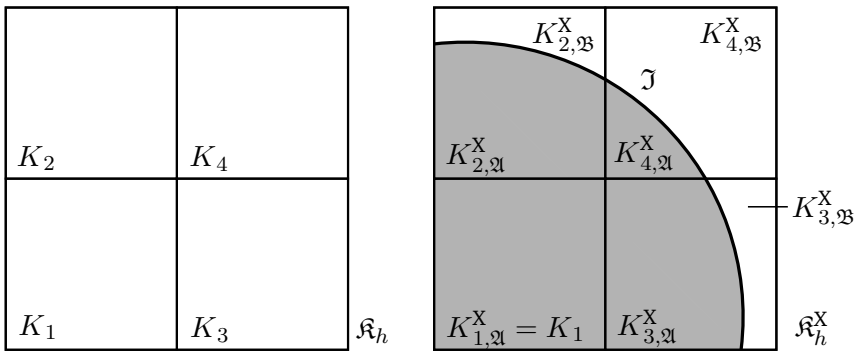


Figure 3.1.: Left: Standard background mesh $\mathfrak{K}_h = \{K_1, K_2, K_3, K_4\}$. Right: Extended cut-cell mesh $\mathfrak{K}^{\mathrm{X}}_h = \{K^{\mathrm{X}}_{1,\mathfrak{A}}, K^{\mathrm{X}}_{2,\mathfrak{A}}, K^{\mathrm{X}}_{2,\mathfrak{B}}, K^{\mathrm{X}}_{3,\mathfrak{A}}, K^{\mathrm{X}}_{3,\mathfrak{B}}, K^{\mathrm{X}}_{4,\mathfrak{A}}, K^{\mathrm{X}}_{4,\mathfrak{B}}\}$ for a given interface $\mathfrak{I}$. Note that $K^{\mathrm{X}}_{1,\mathfrak{A}} = K_1$ and $K^{\mathrm{X}}_{1,\mathfrak{B}} = \emptyset$ (Kummer et al., 2020).

From now on, the 'original cells' K_j will be referred to as background cells. Note, that in background cells where the interface $\mathfrak{I}$ is located, we end up with two cut-cells $K_{j,\mathfrak{A}}(t)$ and $K_{j,\mathfrak{B}}(t)$, see Figure 3.1. For the above defined cut-cell mesh, we extend the basic notations of Definition 1.

Definition 5 (Extended notations). For a polygonal and simply connected computational domain $\Omega = \mathfrak{A}(t) \dot\cup \mathfrak{I}(t) \dot\cup \mathfrak{B}(t) \in \mathbb{R}^D$ ($D \in \{2, 3\}$) we define:

- the numerical time-dependent cut-cell mesh $\mathfrak{K}^{\mathrm{X}}_h(t)$ (Definition 4).
- the set of all edges in the mesh that is extended to $\Gamma := \cup_j \partial K_j \cup \mathfrak{I}$.
- a normal field $\mathbf{n}_\Gamma$ on Γ that is $\mathbf{n}_\Gamma = \mathbf{n}_\mathfrak{I}$ on $\mathfrak{I}$.

The XDG method is essentially a DG method on cut-cells leading to the following definition of the XDG space connecting the standard DG space (Kummer, 2016).

Definition 6 (XDG space). The broken cut-polynomial space of total degree k is defined as:

$$\mathbb{P}^{\mathrm{X}}_k(\mathfrak{K}_h, t) := \big\{ f \in L^2(\Omega); \forall K \in \mathfrak{K}_h : f|_{K \cap \mathfrak{s}(t)} \text{ are polynomial, } \deg(f|_{K\cap\mathfrak{s}(t)}) \le k \big\} = \mathbb{P}_k(\mathfrak{K}^{\mathrm{X}}_h(t)) \tag{3.25}$$

Thus, for cells solely occupied by one phase, i.e. $K_{j,\mathfrak{s}} = K_j$, the standard DG space and the corresponding local approximation are recovered, see Definition 2 and Equation (3.6). In background cells with both phases being present the local approximation of a field property ψ_j is given by

$$\psi_j(\mathbf{x}, t) = \sum_{l=0}^{k} \tilde\psi_{j,l,\mathfrak{A}}(t)\phi_{j,l}(\mathbf{x})1_\mathfrak{A}(\mathbf{x}, t) + \tilde\psi_{j,l,\mathfrak{B}}(t)\phi_{j,l}(\mathbf{x})1_\mathfrak{B}(\mathbf{x}, t), \quad \mathbf{x} \in K_j, \tag{3.26}$$

where $1_\mathfrak{s}(\mathbf{x}, t)$ defines the characteristic function for phase $\mathfrak{s}(t)$, i.e. $1_\mathfrak{s}(\mathbf{x}, t) = 1$ for $\mathbf{x} \in \mathfrak{s}(t)$ and zero everywhere else. Note that the basis functions are not altered from the standard

DG space. The cut-polynomial basis for a phase $\mathfrak{s}(t)$ is given by $\phi^{\mathrm{X}}_{j,l,\mathfrak{s}}(\mathbf{x},t) = \phi_{j,l}(\mathbf{x})1_{\mathfrak{s}}(\mathbf{x},t)$, see Figure 3.2, and the corresponding coefficients are denoted by $\tilde{\psi}_{j,l,\mathfrak{s}}(t)$. Thus, in a single background cell K_j, occupied by both phases, we end up with two separate sets of DOF for each phase.

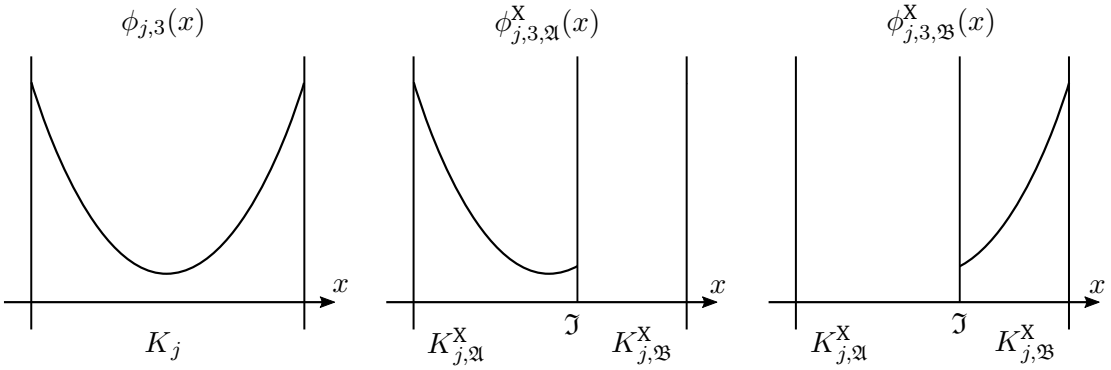


Figure 3.2.: Cut-polynomial basis functions for both phases, i.e. $\phi^{\mathrm{X}}_{j,l,\mathfrak{A}}(\mathbf{x},t)$ and $\phi^{\mathrm{X}}_{j,l,\mathfrak{B}}(\mathbf{x},t)$, in one single 1D background cell. For phase $\mathfrak{A}$ the basis is defined as $\phi^{\mathrm{X}}_{j,l,\mathfrak{A}}(\mathbf{x},t) = \phi_{j,l}(\mathbf{x})1_{\mathfrak{A}}(\mathbf{x},t)$ with $1_{\mathfrak{A}}(\mathbf{x},t) = 1$ for $\mathbf{x} \in \mathfrak{A}(t)$ and $1_{\mathfrak{A}}(\mathbf{x},t) = 0$ for $\mathbf{x} \notin \mathfrak{A}(t)$.

Note that for a high-order representation of the interface $\mathfrak{I}$, resp. level-set field φ, the presented XDG method allows a sub-cell accurate approximation in cut-cells due to the cut-polynomial basis, if combined with a sufficiently accurate numerical integration, see next Section 3.3.2. The formulation of a spatial discretization in the context of an XDG method follows in Section 3.4.

3.3.2. Numerical Integration on Cut-Cells

An essential prerequisite to the XDG method is the high-order numerical integration of cut-cell integrals such as

$$\int_{K^{\mathrm{X}}_{\mathfrak{s}}} f \,\mathrm{d}V, \quad \oint_{\partial K^{\mathrm{X}}_{\mathfrak{s}}} f \,\mathrm{d}S \quad \text{and} \quad \oint_{\mathfrak{I}} f \,\mathrm{d}S, \tag{3.27}$$

which denote integrals over cut-cell volumes $K^{\mathrm{X}}_{\mathfrak{s}} = K \cap \mathfrak{s}$, surfaces $\partial K^{\mathrm{X}}_{\mathfrak{s}} = \partial(K \cap \mathfrak{s})$, and the interface $\mathfrak{I}$. Here, one needs stable and robust integration techniques that allow the accurate integration over domains that are only implicitly defined. In this work two methods are used: The Hierarchical Moment Fitting (HMF) procedure (Müller et al., 2013) and a method proposed by Saye (2015). The HMF quadrature rules support all types of cells such as triangles, quadrilateral, tetra- and hexahedrons. Saye's method is generally faster, but restricted to hyperrectangles.

Hierarchical Moment Fitting

In this section the modified HMF (Gauss and Stokes preserving one-step HMF) following Kummer (2016) is presented. We are considering the numerical integration of (3.27) for some

sufficiently smooth $f \in C^n(K)$ on a reference cell $K \in \mathbb{R}^2$. Thus, we define a quadrature rule

$$\int_{(X,W)}^{\text{num}} f := \sum_{l=1}^{L} w_l f(\mathbf{x}), \tag{3.28}$$

with a given set of nodes $X = \{\mathbf{x}_1, ..., \mathbf{x}_L\}$ and to be determined weights $W = \{w_1, ..., w_L\}$. In the context of XDG we are seeking three sets of weights: $W^{\mathfrak{A}}$ and $W^{\mathfrak{B}}$ for a quadrature over the species volumes and $W^{\mathfrak{I}}$ for the interface surface. Now, for a given set of nodes X and a suitable vector field $\mathbf{f}$ the weights $W^{\mathfrak{A}}$ and $W^{\mathfrak{I}}$ are given as the least-squares solution of the linear system:

$$\int_{(X,W^{\mathfrak{A}})}^{\text{num}} \nabla \cdot \mathbf{f} - \oint_{(X,W^{\mathfrak{I}})}^{\text{num}} \mathbf{f} \cdot \mathbf{n}_{\mathfrak{I}} = \oint_{\partial K^{\mathrm{X}}_{\mathfrak{A}}} \mathbf{f} \cdot \mathbf{n}_{\partial K} \, \mathrm{d}S, \tag{3.29a}$$

$$\oint_{(X,W^{\mathfrak{I}})}^{\text{num}} \kappa \mathbf{n}_{\mathfrak{I}} \cdot \mathbf{f} - (\mathbf{I} - \mathbf{n}_{\mathfrak{I}} \otimes \mathbf{n}_{\mathfrak{I}}) : \nabla \mathbf{f} = - \int_{\partial(\mathfrak{I} \cap K)} \boldsymbol{\tau} \cdot \mathbf{f} \, \mathrm{d}l, \quad \forall \mathbf{f} \in \mathbb{P}_k(\{K\})^2, \tag{3.29b}$$

which consists of the Gauss theorem (3.29a) and the Stokes theorem (3.29b). The general idea is to construct the unknown integrals on the left-hand sides by known exact integrals on the right-hand sides, i.e. $\oint_{\partial K^{\mathrm{X}}_{\mathfrak{A}}} \mathbf{f} \cdot \mathbf{n}_{\partial\Omega} \, \mathrm{d}S$ and $-\int_{\partial(\mathfrak{I} \cap K)} \boldsymbol{\tau} \cdot \mathbf{f} \, \mathrm{d}l$. Therefore, one needs to determine the zero-set of the level-set function φ on the cell boundary ∂K. Note that the number of nodes and weights L is chosen to be 1.6 times the dimension of the function space $\mathbb{P}_k(\{K\})^2$ leading to an underdetermined system with a non-unique solution. Thus, the solution minimizing the l^2-norm is selected. Finally, the quadrature weights $W^{\mathfrak{B}}$ are constructed using the following identity

$$\int_{K^{\mathrm{X}}_{\mathfrak{B}}} f \, \mathrm{d}V = \int_K f \, \mathrm{d}V - \int_{K^{\mathrm{X}}_{\mathfrak{A}}} f \, \mathrm{d}V, \tag{3.30}$$

ensuring that the summation of both cut-cell integrals over $K \cap \mathfrak{A}$ and $K \cap \mathfrak{B}$ equals the integral over K.

Saye's Method

In this section the quadrature method of Saye (2015) is presented. We restrict our description to the two-dimensional case $D = 2$, but note that the method generalizes to arbitrary hyperrectangles in $\mathbb{R}^D$. Like the HMF-quadrature, the quadrature rules for $\mathfrak{A}$ and $\mathfrak{I}$, i.e.

$$\int_{K^{\mathrm{X}}_{\mathfrak{A}}} f \, \mathrm{d}V \approx \sum_l w_l f(x_l) \quad \text{and} \quad \int_{K \cap \mathfrak{I}} g \, \mathrm{d}S \approx \sum_m \hat{w}_m g(\hat{x}_m), \tag{3.31}$$

are constructed recursively until a one-dimensional well-defined integral is integrated by standard numerical methods, e.g. Gaussian quadrature. However, in this case the recursion is performed via a graph of height functions defining the position of the interface $\mathfrak{I}$, see Figure 3.3. The height direction e_i with $i = \{1, 2\}$ for a feasible height function $h(\tilde{x})$ with $\tilde{x} = \{x_1, x_2\} \subseteq K \ominus e_i$ satisfies the condition that $|\partial_i \varphi| > 0$ along $(\tilde{x}, h(\tilde{x}))$. This guarantees that there exists a height function $h(\tilde{x})$ which defines the interface $\mathfrak{I}$ as a graph.

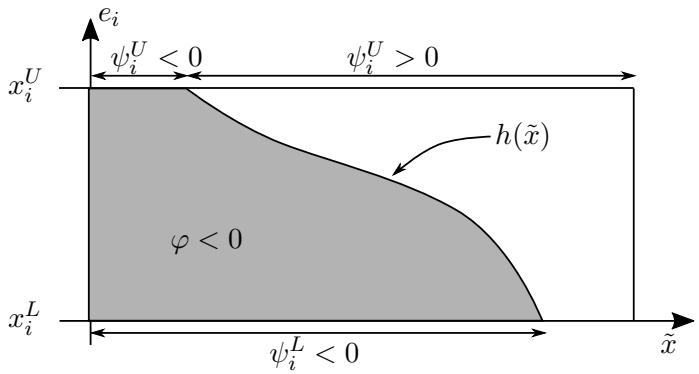


Figure 3.3.: Saye's representation of cut-cells for constructing quadrature rules via height functions (Saye, 2015).

In the end, the volume and surface integrals in (3.31) for a cell $K = (x_1^L, x_1^U) \times (x_2^L, x_2^U) \subset \mathbb{R}^2$ are constructed such that for a selected height direction e_i

$$\int_{K^{\mathfrak{X}}_{\mathfrak{A}}} f = \int_{\hat{\Omega}} \int_{I_i(\tilde{x})} f(\tilde{x} \oplus y e_i) \, \mathrm{d}y \, \mathrm{d}\tilde{x} \quad \text{and} \quad \int_{K \cap \mathfrak{I}} g = \int_{\hat{\Omega}} g \, \frac{|\nabla \varphi|}{|\partial_i \varphi|}\bigg|_{I_i(\tilde{x})} \mathrm{d}\tilde{x}, \tag{3.32}$$

where the inner integral is one-dimensional and well defined. The set $I_i \in \mathbb{R}, i = \{1, 2\}$ is given by

$$I_i(\tilde{x}) = \begin{cases} \{y \in \{x_i^L, x_i^U\} : s\varphi(\tilde{x} + y e_i) > 0\} & \text{if } s = \pm 1, \\ \{y \in \{x_i^L, x_i^U\} : \varphi(\tilde{x} + y e_i) = 0\} & \text{if } s = 0, \end{cases} \tag{3.33}$$

where s denotes a sign indicator of the evaluated domain with

$$s = \Big\{ 1 \text{ for } \mathfrak{A}; \quad 0 \text{ for } \mathfrak{I}; \quad -1 \text{ for } \mathfrak{B}. \tag{3.34}$$

The domain of integration $\tilde{\Omega}$ is defined as the disjoint partitioning $\tilde{\Omega} = V_L \cap V_U$ with

$$V_L = \{\tilde{x} \in K \ominus e_i : s\psi_i^L < 0\} \quad \text{and } V_U = \{\tilde{x} \in K \ominus e_i : s\psi_i^L < 0\} \cup \{\tilde{x} \in K \ominus e_i : s\psi_i^U > 0\}. \tag{3.35}$$

The lower and upper boundaries ψ_i^L and ψ_i^U are given in Figure 3.3. Details on the recursive algorithm implemented in BoSSS are found in Beck (2018).

3.3.3. Cell Agglomeration

The formulation of the XDG method may introduce arbitrarily small cut-cells $K_{j,\mathfrak{s}}$ due to the arbitrary position of the interface. Thus, its volume fraction $|K_{j,\mathfrak{s}}| \,/\, |K_j|$ compared to the background cell $|K_j|$ may become small. This leads to undesirably high condition numbers and further to stability issues, e.g. in the context of a SIP discretization, see Section 3.4.1. Therefore, cell agglomeration is introduced to remove such small cut-cells from the discretized system. Note that the agglomeration is further utilized for the temporal discretization with moving interfaces (Section 3.5) and in the context of an aggregation multi-grid method without a hierarchy of meshes (Kummer et al., 2020).

The cell agglomeration and resulting meshes can be described in terms of graph theory, see Figure 3.4. Considering a numerical mesh $\mathfrak{K}_h^{\mathrm{X}}$, where we denote a general cut-cell by K^{X}, the edge between two cut-cells K_j^{X} and K_g^{X} with $\oint_{\overline{K_j^{\mathrm{X}}} \cap \overline{K_g^{\mathrm{X}}}} 1 \, \mathrm{d}S > 0$ defines a logical edge $Edg(\{K_j^{\mathrm{X}}, K_g^{\mathrm{X}}\})$. Denoting all logical edges in the grid $\mathfrak{K}_h^{\mathrm{X}}$ by $Edg(\mathfrak{K}_h^{\mathrm{X}})$, the pair $\left(\mathfrak{K}_h^{\mathrm{X}}, Edg(\mathfrak{K}_h^{\mathrm{X}})\right)$ defines a undirected graph $G(\mathfrak{K}_h^{\mathrm{X}})$. Thus, logical edges represent the neighbourship between cells within the grid and one can define a cluster of neighbouring cells $a = \{K_1^{\mathrm{X}}, ..., K_L^{\mathrm{X}}\}$ with a corresponding aggregation map $A_a \subset Edg(\mathfrak{K}_h^{\mathrm{X}})$. The aggregation of all cells in a is given by $K_a^{\mathrm{X}} := \cup^{\partial}_{K_l^{\mathrm{X}} \in a} K_l^{\mathrm{X}}$, where the modified union is defined as $X \cup^{\partial} Y := (\overline{X} \cup \overline{Y}) \setminus \partial(\overline{X} \cup \overline{Y})$ in order to ensure simple connectivity of the aggregated cell K_a^{X}. Note that no agglomeration across species is allowed, i.e $Edg(\{K_{l,\mathfrak{A}}, K_{l,\mathfrak{B}}\}) \notin A$.

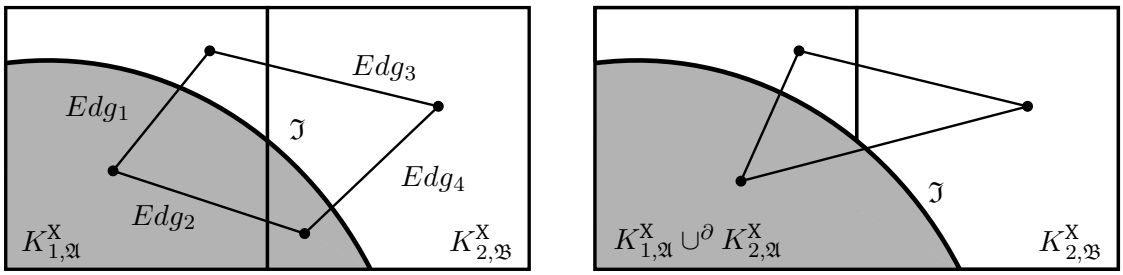


Figure 3.4.: Left: Cut-cell mesh $\mathfrak{K}_h^{\mathrm{X}}$ defined as an undirected graph $G(\mathfrak{K}_h^{\mathrm{X}}) := \left(\mathfrak{K}_h^{\mathrm{X}}, Edg(\mathfrak{K}_h^{\mathrm{X}})\right)$ with logical edges $Edg(\mathfrak{K}_h^{\mathrm{X}}) = \{Edg_1(\{K_{1,\mathfrak{A}}^{\mathrm{X}}, K_{1,\mathfrak{B}}^{\mathrm{X}}\}), Edg_2(\{K_{1,\mathfrak{A}}^{\mathrm{X}}, K_{2,\mathfrak{A}}^{\mathrm{X}}\}), Edg_3(\{K_{1,\mathfrak{B}}^{\mathrm{X}}, K_{2,\mathfrak{B}}^{\mathrm{X}}\}), Edg_4(\{K_{2,\mathfrak{A}}^{\mathrm{X}}, K_{2,\mathfrak{B}}^{\mathrm{X}}\})\}$. Right: Aggregation mesh $Agg(\mathfrak{K}_h^{\mathrm{X}}, A)$ with the Aggregation map $A = \{Edg_2\}$. Note that no agglomeration across different species is allowed.

For a general aggregation map $A \subset Edg(\mathfrak{K}_h^{\mathrm{X}})$ the aggregation mesh $Agg(\mathfrak{K}_h^{\mathrm{X}}, A)$ is the set of all non-aggregated cells and aggregated cells given in A, see Figure 3.4 on the right. The XDG space on such an aggregation mesh is defined as follows:

Definition 7 (Agglomerated XDG space). For some aggregation map $A \subset Edg(\mathfrak{K}_h^{\mathrm{X}})$ and corresponding aggregation mesh $Agg(\mathfrak{K}_h^{\mathrm{X}}, A)$ the agglomerated broken cut-polynomial space of total degree k is defined as:

$$\mathbb{P}_k^{\mathrm{X},A} := \mathbb{P}_k(Agg(\mathfrak{K}_h^{\mathrm{X}}, A)). \tag{3.36}$$

Note that $\mathbb{P}_k(Agg(\mathfrak{K}_h^{\mathrm{X}}, A))$ is a sub-space of the original XDG space $\mathbb{P}_k(\mathfrak{K}_h^{\mathrm{X}})$. A corresponding basis to the aggregation mesh $Agg(\mathfrak{K}_h^{\mathrm{X}}, A)$ is denoted by $\boldsymbol{\phi}^{\mathrm{X},A}$.

As mentioned in the beginning, the cell agglomeration is used to remove small cut-cells from the numerical grid $\mathfrak{K}_h^{\mathrm{X}}$. Therefore, we introduce an agglomeration map $A_\alpha \subset Edg(\mathfrak{K}_h^{\mathrm{X}})$ with edges $Edg(\{K_j^{\mathrm{X}}, K_g^{\mathrm{X}}\})$ matching two conditions:

- The volume fraction of K_j^{X} is smaller than the agglomeration threshold α, i.e $0 < \left|K_j^{\mathrm{X}}\right| / |K_j| < \alpha$.
- The cell K_g^{X} is the neighbour with the largest volume fraction in the same species.

3.4. The Spatial Discretization with the Extended Discontinuous Galerkin Method

In order to formulate the spatial discretization in the context of XDG, we set all integration domains according to the time-level t^{n+1}. So in the following, we are considering a fixed interface at $\mathfrak{I} = \mathfrak{I}(t^{n+1})$ and respectively $\varphi = \varphi(\vec{x}, t^{n+1})$.

3.4.1. The Spatial Discretization for Two-Phase Flow Problems

Before we introduce the discretization of the two-phase Navier-Stokes equations, we first have to define the spaces for the velocity and pressure discretization. Following the work of Kummer (2016) and fulfilling the Ladyženskaja–Babuška–Brezzi condition (see Babuška (1973) and Brezzi (1974)) for the Stokes system with $\rho_{\mathfrak{A}} = \rho_{\mathfrak{B}}$ and $\mu_{\mathfrak{A}} = \mu_{\mathfrak{B}}$, the velocity $\mathbf{u} \in \mathbb{R}^2$ is discretized in an XDG space of order k and the pressure $p \in \mathbb{R}$ with $k' = k - 1$. Thus, we define the function space of ansatz and test functions by

$$\mathbb{V}_{\mathbf{k}}^{\mathrm{X}} := \prod_{k_\gamma}^{\mathbf{k}} \mathbb{P}_{k_\gamma}^{\mathrm{X}}(\mathfrak{K}_h) \tag{3.37}$$

with the degree vector $\mathbf{k} = \{k, k, k-1\} = k_\gamma$ with $\gamma = 1, 2, 3$. Note that there is no rigorous proof of the inf-sup stability for the given $(\mathbf{u}, p)$ pair, which can be directly applied to the presented XDG setting (Kummer, 2016). An inf-sup condition of (3.37) for $k = \{1, 2, 3\}$ is established in Girault et al. (2004) for non-matching grids with triangular finite elements.

Based on the steady spatial discretization given in Kummer (2016), we propose the following discretization for the transient two-phase incompressible Navier-Stokes equations (2.3) with jump conditions (2.16) and (2.17) at the interface $\mathfrak{I}$ and boundary conditions (2.4):
Find $(\mathbf{u}^{n+1}, p^{n+1}) \in \mathbb{V}_{\mathbf{k}}^{\mathrm{X}}$, such that $\forall (\mathbf{v}, q) \in \mathbb{V}_{\mathbf{k}}^{\mathrm{X}}$

$$m(\partial_t \mathbf{u}|_{t^{n+1}}, \mathbf{v}) + c(\mathbf{u}^{n+1}, \mathbf{u}^{n+1}, \mathbf{v}) + b(p^{n+1}, \mathbf{v}) - a(\mathbf{u}^{n+1}, \mathbf{v}) - b(q, \mathbf{u}^{n+1}) = g(\mathbf{v}, q). \tag{3.38}$$

The first bilinear form on the left-hand side corresponds to the time derivative term $\partial_t \mathbf{u}$ and is defined as

$$m(\mathbf{u}, \mathbf{v}) = \int_\Omega \rho \mathbf{u} \cdot \mathbf{v} \, \mathrm{d}V. \tag{3.39}$$

The temporal discretization by the moving interface approach will be discussed in Section 3.5.3. The trilinear form $c(\mathbf{u}^*, \mathbf{u}, \mathbf{v})$ describes the discretization of the convective term, where a local Lax-Friedrichs flux is employed

$$\begin{aligned} c(\mathbf{u}^*, \mathbf{u}, \mathbf{v}) = &- \int_\Omega \rho(\mathbf{u} \otimes \mathbf{u}^*) : \nabla_h \mathbf{u} \, \mathrm{d}V \\ &- \oint_{\Gamma_{\mathrm{int}} \cup \Gamma_{\mathrm{N}} \cup \mathfrak{I}} \left(\{\{\mathbf{u} \otimes \mathbf{u}^*\}\} \, \mathbf{n}_{\mathfrak{I},\Gamma} + \frac{\lambda}{2} [\![\mathbf{u}]\!] \right) \cdot [\![\rho \mathbf{v}]\!] \, \mathrm{d}S. \end{aligned} \tag{3.40}$$

For the choice of the stabilization parameter λ we refer to Kummer (2016). Note that the discretized problem (3.38) needs to be solved iteratively due to the non-linearity of the

convective term. Thus, the whole system is linearized in each iteration with $\mathbf{u}^*$, see Section 3.9.

The bilinear form $b(-,-)$ represents both, the discretization of the pressure gradient and the continuity term, i.e. velocity divergence:

$$b(p,\mathbf{v}) = -\int_\Omega p\nabla_h\cdot\mathbf{v}\ \mathrm{d}V - \oint_{\Gamma_{\text{int}}\cup\Gamma_{\text{D}}\cup\mathfrak{I}} [\![\mathbf{v}]\!]\cdot\mathbf{n}_{\mathfrak{I},\Gamma}\,\{\{p\}\}\ \mathrm{d}S. \tag{3.41}$$

The viscous terms are discretized using an extension to the classical form of the standard symmetric interior penalty method (SIP), first proposed by Arnold (1982). This extension includes the transposed term $\nabla\mathbf{u}^{\mathrm{T}}$ in the divergence operator and since $\nabla_h\mathbf{u}^{\mathrm{T}} : \nabla_h\mathbf{v} = \nabla_h\mathbf{v}^{\mathrm{T}} : \nabla_h\mathbf{u}$, the bilinear form $a(\mathbf{u},\mathbf{v})$ is still symmetric in $\mathbf{u}$ and $\mathbf{v}$

$$\begin{aligned} a(\mathbf{u},\mathbf{v}) = &-\int_\Omega \mu\left(\nabla_h\mathbf{u} : \nabla_h\mathbf{v} + \nabla_h\mathbf{u}^{\mathrm{T}} : \nabla_h\mathbf{v}\right)\ \mathrm{d}V \\ &+ \oint_{\Gamma_{\text{int}}\cup\mathfrak{I}} \left(\left\{\left\{\mu\left(\nabla_h\mathbf{u}+\nabla_h\mathbf{u}^{\mathrm{T}}\right)\right\}\right\}\mathbf{n}_{\mathfrak{I},\Gamma}\right)\cdot[\![\mathbf{v}]\!] \\ &\qquad + \left(\left\{\left\{\mu\left(\nabla_h\mathbf{v}+\nabla_h\mathbf{v}^{\mathrm{T}}\right)\right\}\right\}\mathbf{n}_{\mathfrak{I},\Gamma}\right)\cdot[\![\mathbf{u}]\!]\ \mathrm{d}S \\ &- \oint_{\Gamma_{\text{int}}\cup\mathfrak{I}} \eta\,[\![\mathbf{u}]\!]\cdot[\![\mathbf{v}]\!]\ \mathrm{d}S. \end{aligned} \tag{3.42}$$

In order to ensure coercivity of the form $a(-,-)$ for a suitable norm, the penalty parameter η in the last term has to be chosen according to

$$\eta := \max\{\mu^{\text{in}},\mu^{\text{out}}\}\cdot\max\{\tilde{\eta}^{\text{in}},\tilde{\eta}^{\text{out}}\} \qquad \text{on } \Gamma_{\text{int}} \text{ and} \tag{3.43}$$

$$\eta := \mu^{\text{in}}\cdot\tilde{\eta}^{\text{in}} \qquad \text{on } \partial\Omega, \tag{3.44}$$

where the local penalty factor $\tilde{\eta}$, constant within each cut-cell K^{X}, is computed by

$$\tilde{\eta} = \eta_0 k^2 \frac{\left|\partial K^{\mathrm{X}}\right|}{\left|K^{\mathrm{X}}\right|}. \tag{3.45}$$

In our framework the geometric factor $\left(\left|\partial K^{\mathrm{X}}\right| / \left|K^{\mathrm{X}}\right|\right)$ is already available from the construction of the cut-cell quadrature rules (Section 3.3.2). The safety factor η_0 is chosen to be $\eta_0 = 4.0$ in all simulations in this work. The penalty parameter should be chosen as small as possible, yet large enough to ensure coercivity, since over-penalization increases the condition number and subsequently the approximation error. Further details on the definition of the local penalty factor $\tilde{\eta}$ and the choice of the safety factor η_0 are given in Kummer (2016).

Finally, we specify the term on the right-hand side of the variational formulation (3.38), which summarises the Dirichlet boundary conditions and force terms

$$g(\mathbf{v},q) = t(\mathbf{v}) + s(\mathbf{v}) + r(q). \tag{3.46}$$

The first term describes the Dirichlet boundary conditions of the convective part

$$t(\mathbf{v}) = -\oint_{\Gamma_{\text{D}}} \rho\left((\mathbf{u}_{\text{D}}\otimes\mathbf{u}_{\text{D}})\mathbf{n}_\Gamma + \frac{\lambda}{2}\mathbf{u}_{\text{D}}\right)\cdot\mathbf{v}\ \mathrm{d}S. \tag{3.47}$$

The discretization of the volume term $\mathbf{f}$ and the surface tension force $\sigma\kappa\mathbf{n}_\mathfrak{I}$, see the jump condition (2.31), are given in

$$\begin{aligned} s(\mathbf{v}) = &\int_\Omega \rho\mathbf{f}\cdot\mathbf{v}\,\mathrm{d}V + \oint_\mathfrak{I} \sigma\kappa\mathbf{n}_\mathfrak{I}\cdot\mathbf{v}\,\mathrm{d}S \\ &- \oint_{\Gamma_\mathrm{D}} \mathbf{u}_\mathrm{D}\cdot\left(\nabla_h\mathbf{v}\mathbf{n}_\Gamma + \nabla_h\mathbf{v}^\mathrm{T}\mathbf{n}_\Gamma - \eta\mathbf{v}\right)\,\mathrm{d}S, \end{aligned} \tag{3.48}$$

where the last term corresponds to the Dirichlet boundary condition for the viscous parts. The Dirichlet boundary condition for the continuity equation is defined as

$$r(q) = \oint_{\Gamma_\mathrm{D}} q\mathbf{u}_\mathrm{D}\cdot\mathbf{n}_\Gamma\,\mathrm{d}S. \tag{3.49}$$

3.4.2. Extension of the Spatial Discretization for Contact Line Problems

In order to allow the discretization of the generalized Navier boundary condition (2.25) with (2.28), the surface tension force in Equation (3.48) is discretized via the Laplace-Beltrami formulation. A detailed derivation is given in Section 3.7. Recalling the divergence form (3.106) of the surface tension force on the right-hand side of the momentum equation, the discretization by multiplying the test function $\mathbf{v}$ and integrating by parts over the interface $\mathfrak{I}$ yields

$$\oint_\mathfrak{I} \nabla_\mathfrak{I}\cdot(\sigma\mathbf{P}_\mathfrak{I})\cdot\mathbf{v}\,\mathrm{d}S = -\oint_\mathfrak{I} \sigma\mathbf{P}_\mathfrak{I} : \nabla_\mathfrak{I}\mathbf{v}\,\mathrm{d}S + \int_{\mathfrak{I}\cap\Gamma} \sigma\,\{\{\boldsymbol{\tau}\}\}\cdot[\![\mathbf{v}]\!]\,\mathrm{d}l, \tag{3.50}$$

where $\boldsymbol{\tau}$ is tangential to $\mathfrak{I}$. Note that the curvature κ, which introduces second derivatives of the level-set field φ, see Equation (3.23), does not need to be computed. Further, this formulation introduces integral terms on the cell boundaries only located at the interface, i.e. $\mathfrak{I}\cap\Gamma$. Especially those terms incorporate the contact line L at boundary edges $\Gamma_{\partial\Omega}$. Thus, for the following, we redefine $s(\mathbf{v})$ in Equation (3.48) by replacing the surface tension contribution with the right-hand side of Equation (3.50).

Discretization of the Generalized Navier Boundary Condition According to the variational problem (3.38) in Section 3.4.1, we propose the following discretization based on Reusken et al. (2017) of the contact line problem with the generalized Navier boundary condition (2.25) and (2.28) at the slip wall Γ_S:
Find $(\partial_t\mathbf{u}|_{t^{n+1}}, p^{n+1}) \in \mathbb{V}^\mathrm{X}_\mathbf{k}$, such that $\forall(\mathbf{v}, q) \in \mathbb{V}^\mathrm{X}_\mathbf{k}$

$$m(\mathbf{u}^{n+1},\mathbf{v}) + c(\mathbf{u}^{n+1},\mathbf{u}^{n+1},\mathbf{v}) + \tilde{b}(p^{n+1},\mathbf{v}) - \tilde{a}(\mathbf{u}^{n+1},\mathbf{v}) - \tilde{b}(q,\mathbf{u}^{n+1}) = \tilde{g}(\mathbf{v},q). \tag{3.51}$$

The bilinear form $m(\mathbf{u},\mathbf{v})$ and the trilinear form $c(\mathbf{w},\mathbf{u},\mathbf{v})$ are the same as in the standard two-phase setting. In $\tilde{b}(p,\mathbf{v})$ the boundary term is extended to the domain of the slip wall Γ_S

$$\tilde{b}(p,\mathbf{v}) = -\int_\Omega p\nabla_h\cdot\mathbf{v}\,\mathrm{d}V - \oint_{\Gamma_\mathrm{int}\cup\Gamma_\mathrm{D}\cup\Gamma_\mathrm{S}\cup\mathfrak{I}} [\![\mathbf{v}]\!]\cdot\mathbf{n}_\Gamma\,\{\{p\}\}\,\mathrm{d}S. \tag{3.52}$$

In the viscous terms $\tilde{a}(\mathbf{u}, \mathbf{v})$ both dissipative effective forces $\mathbf{f}_S$ and $\mathbf{f}_L$, defined by the friction coefficients β_S and β_L, are introduced. At the slip wall the SIP form reduces to components in the normal direction

$$\begin{aligned}\tilde{a}(\mathbf{u}, \mathbf{v}) = a(\mathbf{u}, \mathbf{v}) &- \int_{\Gamma_S} \beta_S \mathbf{P}_S \mathbf{u} \cdot \mathbf{P}_S \mathbf{v} \, \mathrm{d}S - \int_L \beta_L (\mathbf{u} \cdot \mathbf{n}_L)(\mathbf{v} \cdot \mathbf{n}_L) \, \mathrm{d}l \\ &+ \int_{\Gamma_S} \mu \left(\mathbf{n}_\Gamma \cdot \left(\nabla_h \mathbf{u} + \nabla_h \mathbf{u}^T \right) \mathbf{n}_\Gamma \right) \cdot (\mathbf{v} \cdot \mathbf{n}_\Gamma) \\ &\qquad + \mu \left(\mathbf{n}_\Gamma \cdot \left(\nabla_h \mathbf{v} + \nabla_h \mathbf{v}^T \right) \mathbf{n}_\Gamma \right) \cdot (\mathbf{u} \cdot \mathbf{n}_\Gamma) \, \mathrm{d}S \\ &- \oint_{\Gamma_S} \eta (\mathbf{u} \cdot \mathbf{n}_\Gamma) \cdot (\mathbf{v} \cdot \mathbf{n}_\Gamma) \, \mathrm{d}S.\end{aligned} \tag{3.53}$$

On the right-hand side of (3.51), i.e $\tilde{g}(\mathbf{v}, q)$, only the term $s(\mathbf{v})$ in (3.46) changes, where the static contact angle θ_{stat} of Young's equation and the current contact angle θ with $\sigma \cos(\theta) \mathbf{n}_L = \sigma \mathbf{P}_S \boldsymbol{\tau}_L$ are summarized

$$\tilde{s}(\mathbf{v}) = s(\mathbf{v}) - \int_L \sigma \cos(\theta_{\text{stat}}) \mathbf{v} \cdot \mathbf{n}_L \, \mathrm{d}l - \int_L \sigma (\mathbf{n}_\Gamma \cdot \mathbf{P}_S \boldsymbol{\tau}_L)(\mathbf{v} \cdot \mathbf{n}_\Gamma) \, \mathrm{d}l. \tag{3.54}$$

3.5. The Temporal Discretization - Coupling the Flow Solver with the Interface Evolution

So far the discretization was exclusively described for the spatial dimension, resulting in a semi-discrete discretization, see Equation (3.38). In this section the temporal discretization for a two-phase problem with moving interface in the context of the XDG method is discussed following Kummer et al. (2018). Therefore, we extended the general conservation law (3.5) by a interface $\mathfrak{I}$ that moves with the interface velocity $w = \mathbf{w} \cdot \mathbf{n}_\mathfrak{I}$. Thus, the initial value problem (IVP) (3.5) additionally needs to fulfil the following Rankine-Hugoniot condition at the interface

$$-w \left[\!\left[\psi \right]\!\right] + \left[\!\left[\mathbf{f}(\psi) \right]\!\right] \cdot \mathbf{n}_\mathfrak{I} = 0. \tag{3.55}$$

In the following we consider one single cut-cell $K_j^{\mathrm{X}}(t)$, where the species index $\mathfrak{s}$ is omitted. The movement of the interface is confined to the corresponding background cell, so there are no topology changes in the cut-cell grid $\mathfrak{K}_h^{\mathrm{X}}(t)$. This issue is addressed in Section 3.5.2. Now, looking at $K_j^{\mathrm{X}}(t)$ the semi-discrete formulation (3.11) for an XDG method with basis $\phi_{j,l}^{\mathrm{X}}(t)$ reads

$$\int_{K_j^{\mathrm{X}}(t)} \frac{\partial \psi_j}{\partial t} \phi_{j,l}^{\mathrm{X}}(t) \, \mathrm{d}V - \int_{K_j^{\mathrm{X}}(t)} \mathbf{f} \cdot \nabla_h \phi_{j,l}^{\mathrm{X}}(t) \, \mathrm{d}V + \oint_{\partial K_j^{\mathrm{X}}(t)} \hat{F}_j \phi_{j,l}^{\mathrm{X}}(t) \, \mathrm{d}S = 0, \quad \forall \phi_{j,l}^{\mathrm{X}}(t). \tag{3.56}$$

Considering the temporal discretization from time step t^0 to t^1, one option is to split the interface evolution from the corresponding partial differential equation (PDE). In this case, first, the interface is moved and second, the discretization is performed for the new fixed interface position. Before performing the discretization, the previous values at t^0 on $K_j^{\mathrm{X}}(t^0)$ need to be extrapolated onto the new cut-cell $K_j^{\mathrm{X}}(t^1)$. Restricted to no topology changes, the

extrapolation operator $\mathrm{Ex}_{(t^0,t^1)}$ is defined as the change of basis for the old coordinates $\tilde{\boldsymbol{\psi}}_j^0$ stated as

$$\begin{aligned} \mathrm{Ex}_{(t^0,t^1)} : \mathbb{P}_k(\mathfrak{K}_h^{\mathrm{X}}(t^0)) &\to \mathbb{P}_k(\mathfrak{K}_h^{\mathrm{X}}(t^1)) \\ \boldsymbol{\phi}_j^{\mathrm{X}}(t^0) \cdot \tilde{\boldsymbol{\psi}}_j^0 &\mapsto \boldsymbol{\phi}_j^{\mathrm{X}}(t^1) \cdot \tilde{\boldsymbol{\psi}}_j^0. \end{aligned} \tag{3.57}$$

Note that $\boldsymbol{\phi}_j^{\mathrm{X}}(t^1) = \boldsymbol{\phi}_j(\mathbf{x}) 1_{\mathfrak{s}}(\mathbf{x}, t^1)$ is essentially an update of the characteristic function. After the extrapolation, the semi-discrete formulation is evaluated for $t = t^1$ and integrated in time over the interval $\left[t^0, t^1\right]$ resulting in

$$\mathbf{M}_j^{\mathrm{X}}(t^1)(\psi_j^1 - \psi_j^0) + \int_{t^0}^{t^1} \left(\oint_{\partial K_j^{\mathrm{X}}(t^1)} \hat{F}_j \phi_j^{\mathrm{X}}(t^1) \, \mathrm{d}S - \int_{K_j^{\mathrm{X}}(t^1)} \mathbf{f} \cdot \nabla \phi_j^{\mathrm{X}}(t^1) \, \mathrm{d}V \right) \mathrm{d}t = 0, \tag{3.58}$$

where the cell-local cut-cell mass matrix is given by

$$\mathbf{M}_j^{\mathrm{X}}(t) = \int_{K_j^{\mathrm{X}}(t)} \boldsymbol{\phi}_j^{\mathrm{X}}(t) \otimes \boldsymbol{\phi}_j^{\mathrm{X}}(t) \, \mathrm{d}V. \tag{3.59}$$

Such a discretization is easy to implement and cheap to compute due to the extrapolation operator (3.57). However, the need to extrapolate limits the coupled scheme to low-order accuracy in time with small time steps (i.e. flow solver and interface evolution). Furthermore, the scheme is non-conservative.

Beside the splitting of the interface evolution from the PDE as described above, there are two more groups of methods. One approach is to perform the time discretization before the spatial discretization leading to products of basis functions at both time steps t^0 and t^1. The second method considers the interface fixed in the space-time domain and discretizes the whole problem simultaneously in space and time. This allows for accuracy in time with large time steps, but at the cost of more globally coupled degrees of freedom and the need of integrating over space-time elements. A more detailed overview on time-stepping schemes for extended methods is found in Kummer et al. (2018) and Fries and Zilian (2009).

3.5.1. Moving Interface Discretization

The moving interface discretization developed by Kummer et al. (2018) allows for high orders of accuracy in time without a full discretization on space-time elements. The basic idea is to evaluate numerical fluxes in a moving reference frame in space and time. On the one hand, this allows to keep the discretization in the spatial dimension and on the other hand, one obtains a conservative scheme in time. Following the work of Kummer et al. (2018), we first rewrite the considered general conservation law (3.5a) in the space-time notation given by

$$\mathrm{div}^* \left((\psi, \mathbf{f})^{\mathrm{T}} \right) = \partial_t \psi + \nabla \cdot \mathbf{f}(\psi) = 0, \tag{3.60}$$

where div^* is denoted as the space-time divergence. In order to formulate the moving interface discretization, a space-time element K_j^* is introduced

$$K_j^* = \{(\mathbf{x}, t) | \mathbf{x} \in K_j^{\mathrm{X}}(t), t^0 < t < t^1\}, \tag{3.61}$$

and the corresponding space-time outer normals on the boundary ∂K_j^* is

$$\mathbf{n}^* = \begin{cases} (1,0)^{\mathrm{T}} & \text{on } K_j^{\mathrm{X}}(t^1) \times \{t^1\}, \\ (-1,0)^{\mathrm{T}} & \text{on } K_j^{\mathrm{X}}(t^0) \times \{t^0\}, \\ \frac{1}{\sqrt{1+w^2}}(-w, \mathbf{n})^{\mathrm{T}} & \text{on } \mathfrak{I}^*, \\ (0, \mathbf{n})^{\mathrm{T}} & \text{elsewhere .} \end{cases} \tag{3.62}$$

In Figure 3.5 a space-time element with the corresponding normal on the boundaries is depicted. Note that in general the cut-cell $K_j^{\mathrm{X}}(t)$ is given for specific species $\mathfrak{s}$, but the index is omitted in this section for readability.

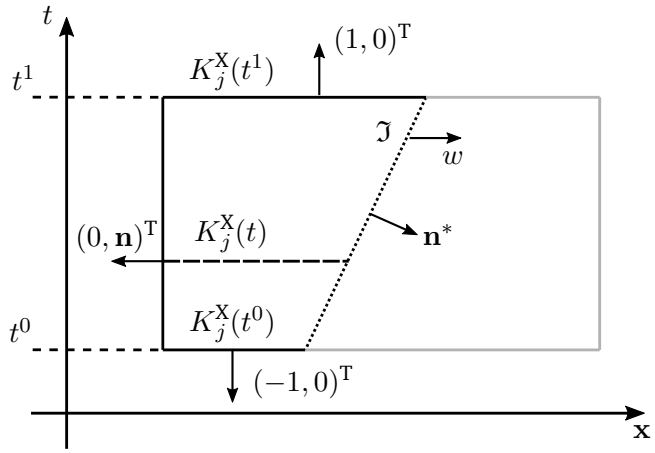


Figure 3.5.: Space-time cut-cell (3.61) and corresponding normal (3.62) on the boundaries (Kummer et al., 2018).

Multiplying Equation (3.60) by the test functions $\boldsymbol{\phi}_{j,l}^{\mathrm{X}}$, integrating over K_j^* and performing partial integration, the DG ansatz for a space-time element reads

$$\begin{aligned} &\int_{K_j^*} \mathrm{div}^* \left((\psi_j, \mathbf{f})^{\mathrm{T}} \right) \boldsymbol{\phi}_{j,l}^{\mathrm{X}} \, \mathrm{d}V^* \\ &= \oint_{\partial K_j^*} (\psi_j, \mathbf{f})^{\mathrm{T}} \cdot \mathbf{n}^* \boldsymbol{\phi}_{j,l}^{\mathrm{X}} \, \mathrm{d}S^* - \int_{K_j^*} (\psi_j, \mathbf{f})^{\mathrm{T}} \cdot \nabla^* \boldsymbol{\phi}_{j,l}^{\mathrm{X}} \, \mathrm{d}V^* = 0. \end{aligned} \tag{3.63}$$

For the numerical fluxes at the boundaries ∂K_j^* the space-time integral identities (13) and (14) in Kummer et al. (2018) are used resulting in

$$\begin{aligned} &\int_{K_j^{\mathrm{X}}(t^1)} (\psi_j \boldsymbol{\phi}_{j,l}^{\mathrm{X}})\big|_{t=t^1} \, \mathrm{d}V - \int_{K_j^{\mathrm{X}}(t^0)} (\psi_j \boldsymbol{\phi}_{j,l}^{\mathrm{X}})\big|_{t=t^0} \, \mathrm{d}V \\ &+ \int_{t^0}^{t^1} \oint_{\partial K_j^{\mathrm{X}}(t)} (-w\psi_j + \mathbf{f} \cdot \mathbf{n}) \phi_j^{\mathrm{X}}(t) \, \mathrm{d}S \, \mathrm{d}t - \int_{t^0}^{t^1} \int_{K_j^{\mathrm{X}}(t)} \mathbf{f} \cdot \nabla \phi_j^{\mathrm{X}}(t) \, \mathrm{d}V \, \mathrm{d}t. \end{aligned} \tag{3.64}$$

Note that the numerical flux $\hat{F}_j^* \approx -w\psi_j + \mathbf{f} \cdot \mathbf{n}$ on the static parts of the boundary $\partial K_j^* \setminus \mathfrak{I}(t)$ reduces to $\hat{F}_j^* = \hat{F}_j \approx \mathbf{f} \cdot \mathbf{n}$. Furthermore, the first two integrals on the respective time step t^0

and t^1 may be expressed in terms of the corresponding cut-cell mass matrix. Thus, the moving interface discretization reads

$$\mathbf{M}_j^{\mathrm{X}}(t^1)\psi_j^1 - \mathbf{M}_j^{\mathrm{X}}(t^0)\psi_j^0 + \int_{t^0}^{t^1} \left(\oint_{\partial K_j^{\mathrm{X}}(t)} \hat{F}_j^* \phi_j^{\mathrm{X}}(t) \, \mathrm{d}S - \int_{K_j^{\mathrm{X}}(t)} \mathbf{f} \cdot \nabla \phi_j^{\mathrm{X}}(t) \, \mathrm{d}V \right) \mathrm{d}t = 0. \quad (3.65)$$

Comparing with the splitting approach, the moving interface discretization additionally incorporates the mass matrix of the previous time step and the spatial integrals are now time-dependent. The numerical flux $\hat{F}_j^*$ on the interface boundary $\mathfrak{I}^*$ is still to be defined. Here, a natural requirement is the free-stream preservation, which demands an unaltered solution for an interface moving through an initially constant solution. Therefore, two options are possible to define $\hat{F}_j^*$: either adding the moving interface correction term $-w\psi$ to the standard static flux $\hat{F}_j$, or actually considering the flux $\hat{F}_j^*$ in a moving frame. For the first option we set

$$\hat{F}_j^* = -w\psi^{\mathrm{dw}} + \hat{F}_j, \quad (3.66)$$

where the downwind value ψ^{dw} needs to be chosen due to causality reason. This means that new values should only depend on previous ones. For the second option one needs to consider the specific underlying equation. For example taking the scalar convection with the linear flux $\mathbf{f}(\psi) = \mathbf{c}\psi$, the moving interface flux is given by the upwind formulation

$$\hat{F}_j^* = \begin{cases} (-w\mathbf{n} + \mathbf{c}) \cdot \mathbf{n}\psi^{\mathrm{in}} & \text{if } -\psi + \mathbf{c} \cdot \mathbf{n} \geq 0, \\ (-w\mathbf{n} + \mathbf{c}) \cdot \mathbf{n}\psi^{\mathrm{out}} & \text{if } -\psi + \mathbf{c} \cdot \mathbf{n} < 0. \end{cases} \quad (3.67)$$

3.5.2. Topology Changes

So far the movement of the interface has been confined to one background cell. Now, the cut-cell mesh topology may change in time leading to appearing new cut-cells and disappearing cut-cells in $\mathfrak{K}_h^{\mathrm{X}}$. This issue is handled by cell agglomeration introduced in Section 3.3.3.

Cell Agglomeration for the Splitting Approach Considering the splitting approach, the extrapolation operator needs to be extended for appearing cells, since there are no coordinates at the previous time step in the new cut-cell. Thus, the agglomeration map A in Section 3.3.3 is extended by logical edges $Edg(\{K_j^{\mathrm{X}}, K_g^{\mathrm{X}}\})$ matching the following conditions:

- A cut-cell appears: $\left|K_j^{\mathrm{X}}(t^0)\right| = 0$ and $\left|K_j^{\mathrm{X}}(t^1)\right| > 0$
- The cell K_g^{X} is the neighbour with the largest volume fraction in the same species.

As a result, the extrapolation operator extends to

$$\begin{aligned} \mathrm{Ex}_{(t^0,t^1)} : \mathbb{P}_k(Agg(\mathfrak{K}_h^{\mathrm{X}}, A^0)) &\to \mathbb{P}_k(Agg(\mathfrak{K}_h^{\mathrm{X}}, A^1)) \\ \boldsymbol{\phi}^{\mathrm{X},A^0}(t^0) \cdot \tilde{\boldsymbol{\psi}}^0 &\mapsto \boldsymbol{\phi}^{\mathrm{X},A^1}(t^1) \cdot \tilde{\boldsymbol{\psi}}^0, \end{aligned} \quad (3.68)$$

where A^1 includes the logical edge described above, whereas A^0 not.

Cell Agglomeration for the Moving Interface Approach Since the moving interface discretization includes both cut-cell mass matrices $\mathbf{M}^{\mathrm{X}}(t^0)$ and $\mathbf{M}^{\mathrm{X}}(t^1)$, also disappearing cut-cells need to be handled. Thus, the agglomeration map A further extends to logical edges $Edg(\{K_j^{\mathrm{X}}, K_g^{\mathrm{X}}\})$ matching the following conditions:

- A cut-cell disappears: $\left|K_j^{\mathrm{X}}(t^0)\right| > 0$ and $\left|K_j^{\mathrm{X}}(t^1)\right| = 0$
- The cell K_g^{X} is the neighbour with the largest volume fraction in the same species.

The situation of a disappearing and a new appearing cut-cell is depicted in Figure 3.6.

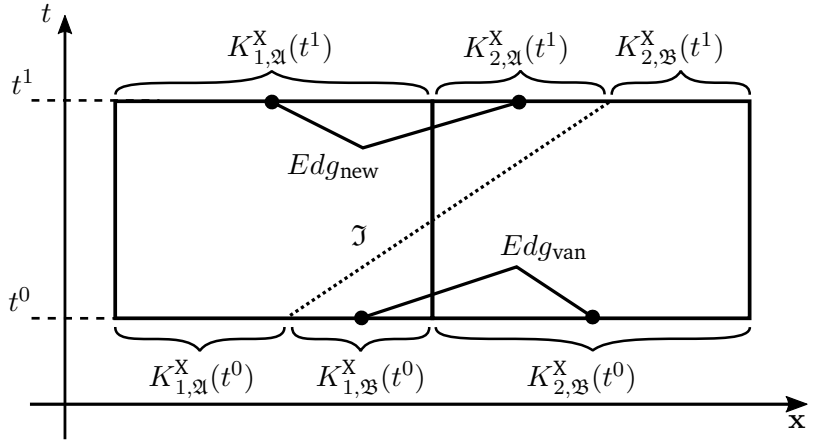


Figure 3.6.: In case of the interface entering a new cell during a time step, new cut-cells appear $K^{\mathrm{X}}_{2,\mathfrak{A}}(t^1)$ and old cells $K^{\mathrm{X}}_{1,\mathfrak{B}}(t^0)$ vanish. Accordingly, the agglomeration maps for both time levels read: $A(t^0) = \{Edg_{\mathrm{van}}\}$ and $A(t^1) = \{Edg_{\mathrm{new}}\}$.

3.5.3. The Temporal Discretization for Two-Phase Navier-Stokes Equations

Incorporating the moving interface approach for the temporal discretization of the transient Navier-Stokes equations (3.38) one can show that the numerical flux $\hat{F}_j^* \approx -w\psi_j + \mathbf{f}\cdot\mathbf{n}$ at the moving frame cancels out identically with the contribution of the convective term. In context of the variational formulation (3.38) the numerical flux $\hat{F}_j^*$ reads

$$\hat{F}_j^* = -w\mathbf{u}_j + (\mathbf{u}_j \otimes \mathbf{u}_j)\cdot\mathbf{n}_{\mathfrak{I}} \tag{3.69}$$

and since considering a material interface with $w = \mathbf{u}_{\mathfrak{I}}\cdot\mathbf{n}_{\mathfrak{I}}$ both terms cancel out

$$\hat{F}_j^* = -\left(\mathbf{u}_j\cdot\mathbf{n}_{\mathfrak{I}}\right)\mathbf{u}_j + \left(\mathbf{u}_j\cdot\mathbf{n}_{\mathfrak{I}}\right)\mathbf{u}_j = 0. \tag{3.70}$$

Thus for the moving interface discretization, the contribution of the convective term at the interface $\mathfrak{I}$ is removed in (3.40).

For the discretization of the temporal term $m(\partial_t\mathbf{u}^{n+1}, \mathbf{v})$ in (3.38), we use standard backward differentiation formula (BDF) schemes of order $l = \{1, 2, 3\}$ with

$$m(\mathbf{u}^{n+1}, \mathbf{v}) - \Delta t b_0 NSE\left((\mathbf{u}^{n+1}, p^{n+1}), (\mathbf{v}, q)\right) = \sum_{j=1}^{l} a_j m(\mathbf{u}^{n+1}, \mathbf{v}), \tag{3.71}$$

where the corresponding coefficients to each order are given in Table 3.1 according to Deville et al. (2002). The linear form $NSE(-,-)$ denotes the Navier-Stokes operator including $c(-,-,-)$, $b(-,-)$, $a(-,-)$ and the right-hand side $g(-,-)$.

Table 3.1.: Coefficients of the l-step BDF scheme (3.71) for $l \leq 3$. For a given l one has $b_0 = \frac{\gamma_0}{\underline{\gamma}}$ and $a_s = \frac{\gamma_s}{\underline{\gamma}}$ with $1 \leq s \leq l$ (Deville et al., 2002).

l	$\underline{\gamma}$	γ_0	γ_1	γ_2	
1	1	1	1	-	-
2	3	2	4	-1	-
3	11	6	18	-9	2

The BDF schemes are implicit multistep methods and thus, are A-stable for $l \leq 2$ (Dahlquist, 1963). For higher order schemes the stability region includes more eigenvalues z with $\Re(z) < 0$. However, note that the unstable eigenvalues for $l = 3$ are comparable small, i.e. near the imaginary axis. Furthermore, one should note that according to Kummer et al. (2018) a spatial approximation of degree k theoretically requires a time integration scheme of at least $2k$. Since in the context of multi-phase flows we are dealing with comparably small time steps due to the capillary time step restriction (3.119), we also allow the use of a third order BDF scheme.

3.6. Level-Set Representation and Evolution of the Interface

So far the level-set function φ was not further defined, expect that the sign need to correspond to the respective domain $\mathfrak{A}$ or $\mathfrak{B}$, see Equation (3.21). In order to effectively evaluate the zero-set of φ, i.e. the interface $\mathfrak{I}$, and the corresponding normal field $\mathbf{n}_\mathfrak{I}$, φ is in general defined as a signed-distance function

$$\varphi(\mathbf{x}) = \begin{cases} -\text{dist}(\mathbf{x}, \mathfrak{I}), & \mathbf{x} \in \mathfrak{A}, \\ \text{dist}(\mathbf{x}, \mathfrak{I}), & \mathbf{x} \in \mathfrak{B}, \\ 0, & \mathbf{x} \in \mathfrak{I}, \end{cases} \tag{3.72}$$

where $\text{dist}(\mathbf{x}, \mathfrak{I})$ denotes the shortest distance of $\mathbf{x} \in \Omega$ to the interface $\mathfrak{I}$. One important property of this definition is that $|\nabla\varphi| = 1$. Thus, Equation (3.72) can be rewritten in terms of the Eikonal equation with

$$\begin{aligned} |\nabla\varphi| - 1 &= 0 \quad \text{in } \Omega, \\ \varphi &= 0 \quad \text{on } \mathfrak{I}. \end{aligned} \tag{3.73}$$

Considering the discretization of the signed-distance level-set function, we approximate $\varphi = \varphi^{\text{dg}}$ by a standard DG field in $\mathbb{P}_k(\mathfrak{K}_h)$. This allows a high-order representation of the interface with sub-cell accuracy (Section 3.3.1). However, the approximation in $\mathbb{P}_k(\mathfrak{K}_h)$ inherently introduces jumps at the cell boundaries. Thus, the assumption of an at least $C^0(\Omega)$-continuous interface is not given. Therefore, we introduce a second level-set approximation $\varphi^{\text{c0}} \in \mathbb{P}_k(\mathfrak{K}_h) \cap C^0(\Omega)$, that is constructed via a L^2-projection with continuity constraints at the cell boundaries, see Section 3.6.1. This level-set is exclusively used for the definition

of the computational domain $\Omega = \mathfrak{A}(t) \dot{\cup} \mathfrak{I}(t) \dot{\cup} \mathfrak{B}(t)$ and its normal field $\mathbf{n}_\mathfrak{I}$ for the spatial discretization in Section 3.3.

The level-set φ^{dg} is used for the evolution of the interface. Recalling Section 2.2, the evolution of the interface described by an implicit function is given by Equation (2.14). Considering a material interface, the bulk velocity $\mathbf{u}$ can be used to advect φ^{dg} on the computational domain Ω. However, this would not preserve the desired signed-distance property $|\nabla\varphi| = 1$. One should note that for the advection equation (2.14) the only valid velocity for the evolution should be the one at the interface itself, i.e. $\mathbf{w} = \mathbf{u}_\mathfrak{I}$. In order to advect the entire level-set field in Ω, one needs to extend the interface velocity $\mathbf{u}_\mathfrak{I}$ onto Ω. This extension problem and the construction of a extension-velocity field $\mathbf{u}_{\text{ext}}$ will be discussed in Section 3.6.3.

Since we are dealing with a sharp interface representation in the context of an XDG method, the approximation space for the bulk velocity is in ${\mathbb{P}_k}^{\text{X}}(\mathfrak{K}_h)$, whereas $\varphi^{\text{dg}} \in \mathbb{P}_k(\mathfrak{K}_h)$. In order to generate a velocity field for the advection equation, we introduce a density-averaged velocity field in $\mathbb{P}_k(\mathfrak{K}_h)$ with

$$\mathbf{u}_{\rho\text{Aver}} = \frac{\rho_\mathfrak{A}\mathbf{u}_\mathfrak{A} + \rho_\mathfrak{B}\mathbf{u}_\mathfrak{B}}{\rho_\mathfrak{A} + \rho_\mathfrak{A}}. \tag{3.74}$$

Note that both φ^{dg} and $\mathbf{u}_{\rho\text{Aver}}$ have the same polynomial degree k.

Even when advecting the level-set field by an extension-velocity field $\mathbf{u}_{\text{ext}}$, the signed-distance property may be distorted. Reason for that are numerical diffusion and an inaccurate approximation of the signed-distance function by polynomials. Thus, it is necessary to reinitialize the level-set field in every time step. Reducing the computational costs the reinitialization might be carried out after a series of time steps. The reinitialization techniques used in this work are presented in Section 3.6.2.

So far the level-set function and corresponding methods and operations were considered on the computational domain Ω. However, the zero-set is actually the only physical meaningful part of the whole level-set function. Thus the description of the signed-distance function may be restricted to a small area around the zero-set, denoted as the narrow-band, see Figure 3.7.

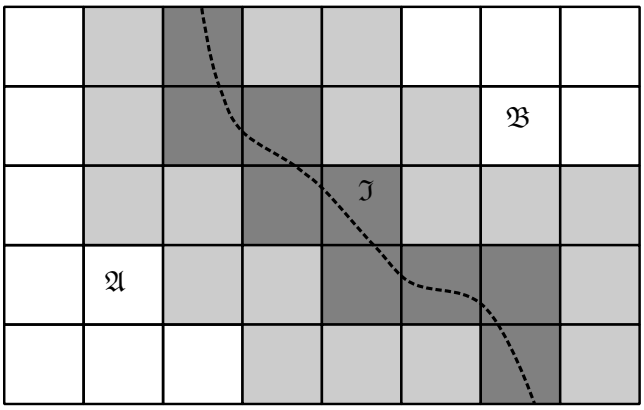


Figure 3.7.: The narrow-band around the interface includes cut-cells $\mathfrak{K}_{\text{cc}}$ (dark grey cells) and the neighbours $\mathfrak{K}_{\text{near}}$ (light grey) considering the vertices of the cut-cells. The white cells are denoted as far-field cells $\mathfrak{K}_{\text{far}}$.

It is sufficient to define the signed-distance function on the cut-cells and its neighbours denoted as $\mathfrak{K}_{\text{cc}}$ and $\mathfrak{K}_{\text{near}}$, respectively. Outside this narrow-band, i.e. denoted as far-field $\mathfrak{K}_{\text{far}}$, the level-set function is either $\varphi^{\text{dg}} = -1$ in $\mathfrak{A}$ and respectively $\varphi^{\text{dg}} = +1$ in $\mathfrak{B}$. For φ^{c0} the significant cells are only the cut-cells. With this reduction of valid cells for the signed-distance function, one is able to introduce effectively fast-marching algorithms, where the above operations are performed cell-wise from the interface to the far-field. A level-set evolution algorithm based on the fast-marching procedure is presented in Section 3.6.4.

Recalling Section 2.2 once again, there is a second description of the interface: The explicit parametric formulation. In this work such an interface representation was developed based on the Fourier transformation, which allows an exact computation of the surface tension force. However, this representation is only applicable to wave and circle-like interfaces. This Fourier-based level-set is described in Section 3.6.5.

3.6.1. Continuity Projection of the Interface

DG fields are in general discontinuous between cells. However, the interface respectively the level-set field should be continuous. Therefore, a L^2-projection with continuity constrains was implemented.

Consider a given DG field $\psi \in \mathbb{P}_k(\mathfrak{K}_h)$, which is in general discontinuous over cell boundaries. In order to enforce continuity along cell edges, it is sufficient to enforce continuity just at some points on the internal edges Γ_{int}. For example taking a polynomial basis in $\mathbb{P}_k^2$, the cell boundary (spatial dimension reduced by 1) is consequently described by a polynomial $\mathbb{P}_k^1$. Thus the interpolation of sufficient points at the edge with equality constraints, leads to continuity of the whole polynomial representation in $\mathfrak{K}_h$.

Our objective is to find an optimal L^2-projection of $\varphi^{\text{dg}} \in \mathbb{P}_k$ onto $\varphi^{\text{c0}} \in \mathbb{P}_{k^*}, k^* \geq k$, such that $\varphi^{\text{c0}} \in C^0$. This can be described as a quadratic optimization problem (OP) with equality constraints at the internal edges Γ_{int}:

$$\min \left\| \varphi^{\text{c0}} - \varphi^{\text{dg}} \right\|_2^2 \quad \text{on } \mathfrak{K}_h \tag{3.75a}$$

$$\text{s.t.} \quad \varphi^{\text{c}}_{j\text{in}}|_{\Gamma_i} = \varphi^{\text{c}}_{j\text{out}}|_{\Gamma_i}, \quad \forall \Gamma_i \tag{3.75b}$$

Let be $\varphi^{\text{c0}} = \mathbf{x}^{\text{T}}\boldsymbol{\phi}$ with $\mathbf{x}$ the sought-after coordinate vector and $\varphi^{\text{dg}_*} = \mathbf{b}^{\text{T}}\boldsymbol{\phi}$ the projection of φ^{dg} onto $\mathbb{P}_{k^*}$. Thus, the OP can be equivalently stated as follows

$$J(\mathbf{x}) = \frac{1}{2}\mathbf{x}^{\text{T}}\mathbf{M}\mathbf{x} - \mathbf{x}^{\text{T}}\mathbf{b} + d \rightarrow \min \tag{3.76a}$$

$$\text{s.t.} \quad \mathbf{A}\mathbf{x} = 0 \tag{3.76b}$$

where $\mathbf{M} = \mathbf{I} \in \mathbb{R}^{N_{k^*} \times N_{k^*}}$, since $\boldsymbol{\phi}$ is a orthonormal basis, $d = \mathbf{b}^{\text{T}}\mathbf{b}$ and $\mathbf{A} \in \mathbb{R}^{N_{\text{c}} \times N_{k^*}}$ describes the constraint matrix with N_{c} denoting the number of constraints at the internal edges.

In order to incorporate the equality constraints we follow the Karush-Kuhn-Tucker conditions of first order. Therefore, the objective function $J(\mathbf{x})$ is extended by the Lagrange multiplier

$\boldsymbol{\lambda} \in \mathbb{R}^{N_c}$ that satisfy the equality constraints:

$$L(\mathbf{x}, \boldsymbol{\lambda}) = \frac{1}{2}\mathbf{x}^T\mathbf{M}\mathbf{x} - \mathbf{x}^T\mathbf{b} + d + \boldsymbol{\lambda}^T\mathbf{A}\mathbf{x} \rightarrow \min \tag{3.77}$$

Thus the solution to the OP can be described as the solution of the following system of linear equations

$$\mathbf{M}\mathbf{x} - \mathbf{b} + \mathbf{A}^T\boldsymbol{\lambda} = 0, \tag{3.78a}$$

$$\mathbf{A}\mathbf{x} = 0. \tag{3.78b}$$

Pre-multiplying both sides of equation (3.78a) by $\mathbf{A}\mathbf{M}^{-1}$, we get

$$\mathbf{A}\mathbf{x} + \mathbf{A}\mathbf{M}^{-1}\mathbf{A}^T\boldsymbol{\lambda} = \mathbf{A}\mathbf{M}^{-1}\mathbf{b}, \tag{3.79}$$

where $\mathbf{A}\mathbf{M}^{-1}\mathbf{A}^T = S \in \mathbb{R}^{N_c \times N_c}$ is the Schur complement of the OP matrix $\mathbf{M}$. Since $\mathbf{M} = \mathbf{I}$ is a positive definite symmetric matrix, the Schur complement $S = \mathbf{A}\mathbf{A}^T$ is a symmetric positive definite matrix. Using the equality constraint (3.78b), Equation (3.79) reduces to

$$\mathbf{A}\mathbf{A}^T\boldsymbol{\lambda} = \mathbf{A}\mathbf{b}. \tag{3.80}$$

Thus, the solution of (3.78) is reduced to solving (3.80), since the solution vector $\mathbf{x}$ is given by the evaluation of (3.78a) with

$$\mathbf{x} = \mathbf{b} - \mathbf{A}^T\boldsymbol{\lambda}. \tag{3.81}$$

Note that the problem size is reduced to the number of the constraint conditions N_c. The size is even further reduced, when only the projection on the narrow-band, i.e. cut-cells and neighbouring cells, is considered. Thus, a direct solver is used for Equation (3.80).

Structure and Construction of the Constraint Matrix $\mathbf{A}$ Consider the continuity projection of $\varphi^{dg} \in \mathbb{P}_k$ with $k = 1$ onto $\varphi^{c0} \in \mathbb{P}_k$ on the domain $\Omega = \{K_1, K_2, K_3, K_4\}$ as depicted in Figure 3.8.

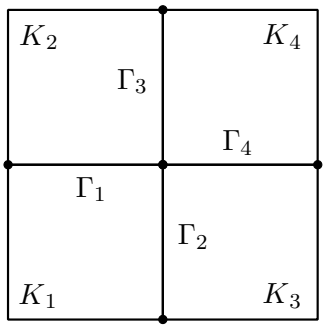


Figure 3.8.: Ensuring continuity for $k = 1$ between all cells $\{K_1, K_2, K_3, K_4\}$, it is sufficient to enforce continuity between two adjacent cells at two points on the internal edges $\{\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4\}$

In order to ensure continuity along all internal edges $\Gamma_{\text{int}} = \{\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4\}$ it is sufficient for $k = 1$ to ensure continuity at two points on each edge, since the polynomial representation

along each edge is linear and thus exactly determined by two points. For $k = 1$ we take the end points of the corresponding edge, see Figure 3.8. Enforcing continuity for $\varphi^{c0} = \mathbf{x}^{\mathrm{T}}\phi = \sum_j \sum_n x_{j,n}\phi_{j,n}$ at one point pt we require that

$$\sum_n x_{j1,n}\phi_{j1,n}(pt) - \sum_n x_{j2,n}\phi_{j2,n}(pt) = 0, \tag{3.82}$$

where $j1$ and $j2$ denote the cell indices of the adjacent cells at the corresponding edge. This condition (3.82) represents one row in the constraint matrix $\mathbf{A}$. Thus, the full matrix is constructed by enforcing condition (3.82) on each point. However, one needs to take care of redundant conditions. Looking at Figure 3.8, it is obvious that enforcing continuity on three edges, the continuity of the fourth edge is implicitly enforced in the point where all four edges are connected. Thus, only one more condition at the other point needs to be set resulting in a total number of seven conditions for the shown example. Note that for this example the continuity projection of a DG field with twelve coordinates is solved via a linear system (3.80) with dimension of $\mathbb{R}^{7\times 7}$.

This continuity projection technique is also applicable for domains with hanging nodes, only to be extended for the number of independent conditions at the hanging nodes, i.e. there are only three connecting edges in 2D. For the extension to 3D the reasoning for edges is also given for faces. In this case one needs a sufficient number of points in order to define a plane with the corresponding polynomial order. Note that for the example above the extension into the third dimension results implicitly in an already continuous edge. Furthermore, the local construction of the constraint matrix allows the use in multi-core computations with low inter-process communication. In this case, the continuity constraints only at inter-process edges need to be exchanged.

3.6.2. Reinitialization

In order to maintain the signed-distance property during the simulation with multiple advection steps, the level-set function needs to be reinitialized. In general, the reinitialization methods can be divided into two main groups: Geometry-based methods aiming to directly solve the signed-distance property (3.72), or PDE-based approaches solving the Eikonal equation (3.73).

The first group includes methods, which for every quadrature point in the computational domain determine the closest point on the interface (Marchandise et al., 2007; Saye, 2015). Further, methods that construct iteratively the level-set field point by point or cell-wise starting from the interface (see Adalsteinsson and Sethian (1995) for the original fast-marching algorithm). Such iterative methods may additionally make use of Gauss-Seidel iterations known as fast sweeping methods (Zhao, 2004).

PDE-based approaches globally solve the Eikonal equation (3.73) or variants of it on the whole computational domain. The reformulations include hyperbolic (Sussman et al., 1994), parabolic (Chunming Li et al., 2005) and elliptic (Basting and Kuzmin, 2013) approaches. The presented method below is based on a elliptic reformulation.

Elliptic Reinitialization

The elliptic reinitialization is presented according to the work of Utz et al. (2017), based on the idea of Basting and Kuzmin (2013). The following minimization problem of the energy functional $E(\varphi)$ is considered

$$E(\varphi) = \int_\Omega \psi(|\nabla\varphi|)\ \mathrm{d}V \rightarrow \min \quad \text{in } \Omega \tag{3.83a}$$

$$\text{s.t. } \varphi = 0 \quad \text{on } \mathfrak{I} \tag{3.83b}$$

with a potential function $\psi(s)$. There are two options for the potential: The single-well potential $\psi_1(s) = \frac{1}{2}(s-1)^2$ with the solution $s = |\nabla\varphi| = 1$ and the double-well potential

$$\psi_2(s) = \begin{cases} \frac{1}{2}s^2(s-1)^2 & \text{for } s \leq 1 \\ \frac{1}{2}(s-1)^2 & \text{for } s > 1. \end{cases} \tag{3.84}$$

with an additional solution at $s = 0$. The double-well potential is introduced in order to circumvent numerical instabilities, which arise for the single-well potential $\psi_1(s)$ in regions, where $|\nabla\varphi| \rightarrow 0$. However, $\psi_2(s)$ may flatten the level-set function in such regions. Therefore, a three-step preconditioning algorithm was presented in Utz (2018).

The minimization problem (3.83) can be equivalently rewritten into the following elliptic form

$$\begin{aligned} \frac{\partial E}{\partial \varphi} = \nabla \cdot (d(|\nabla\varphi|)\nabla\varphi) = 0 \quad & \text{in } \Omega \\ \varphi = 0 \quad & \text{on } \mathfrak{I}, \end{aligned} \tag{3.85}$$

where the associated diffusion rate is given by $d(s) = \frac{\mathrm{d}\psi(s)}{\mathrm{d}s}\frac{1}{s}$. For the DG discretization and the choice of the penalty parameter for the minimization constraints at the interface, the reader is referred to Utz (2018). The resulting non-linear system is solved using a fix-point iteration.

Fast-Marching Reinitialization

The basic idea of the fast-marching algorithm is to iteratively construct a field starting from a subset of points with some initial information in such a way that the information propagates from points close to the initial set to more distant points. In the context of a reinitialization algorithm, the level-set field is constructed from the interface point-wise to the far field. Following the work of Adalsteinsson and Sethian (1995), the points in a grid are divided into three sets: *Accepted*, *Close* and *Far*. Points in *Accepted* are already solutions of the Eikonal equation (3.73) and the neighbouring points are denoted as *Close*. All other point are in *Far*. The marching algorithm 1 recalculates the solution of (3.73) for each *Close* point in every iteration. The point with the smallest value moves from this set to *Accepted* and all neighbouring cells to *Close*. This procedure repeats until all points are *Accepted*.

In the context of a DG method with a polynomial approximation of φ in each cell K, the algorithm is processed on two different levels of the grid. On the upper level the points

Algorithm 1 Fast-marching reinitialization

procedure FASTMARCHREINIT(φ^0, pointsI)
  Accepted ←pointsI
  Close ←Neighbour(pointsI)
  Far ←points $\notin$ *Accepted* $\cup$ *Close*
  while *Close* $\neq \emptyset$ **do**
    $\varphi^*_{\text{pt}} \leftarrow$LocalSolve($\varphi^0_{\text{pt}}$) $\forall$ points $\in$ *Close* ▷ Solve (3.73)
    Accepted ←point($\varphi^*_{\text{pt,min}}$) ▷ Remove point from *Close*
    Close ←Neighbour(point($\varphi^*_{\text{pt,min}}$)) ▷ Remove point from *Far*
  end while
  return φ^*
end procedure

represent the cells in the numerical grid $\mathfrak{K}_h$. On the lower level each cell consist of local quadrature nodes. Thus, a call of LocalSolve() on the grid level represents a local fast-marching procedure on the cell level. Since a polynomial basis is considered in this work, the solution on the quadrature nodes needs to be projected onto the polynomial basis after it is accepted.

3.6.3. Construction of an Extension-Velocity

In order to preserve the signed-distance property and reducing the need for reinitialization, the construction of an extension-velocity field is used. As mentioned above, the only relevant velocity for the evolution of the level-set function is the velocity at the interface $\mathbf{u}_\mathfrak{I}$. Thus, one wants to construct a velocity field $\mathbf{u}_{\text{ext}}(\mathbf{x})$, such that

$$\mathbf{u}_{\text{ext}}(\mathbf{x}) = \mathbf{u}_\mathfrak{I}(\mathbf{x}_{\text{cp}}(\mathbf{x})), \tag{3.86}$$

where $\mathbf{x}_{\text{cp}}(\mathbf{x})$ defines for $\mathbf{x} \in \Omega$ the closest point on the interface by

$$\mathbf{x}_{\text{cp}}(\mathbf{x}) = \arg\min_{\mathbf{x}_\mathfrak{I} \in \mathfrak{I}} (|\mathbf{x}_\mathfrak{I} - \mathbf{x}|). \tag{3.87}$$

For a signed-distance level-set function such a construction of an extension-velocity field can be equivalently described as the solution of the following PDE for each velocity component u^d with $d = \{1, 2\}$:

$$\nabla\varphi \cdot \nabla u^d = 0 \quad \text{in } \Omega, \tag{3.88a}$$

$$u^d = u^d_\mathfrak{I} \quad \text{on } \mathfrak{I}. \tag{3.88b}$$

Since the solution $\mathbf{u} = \mathbf{u}_\mathfrak{I}$ propagates component-wise from the interface along characteristics perpendicular to the interface, the problem is not well-posed for interfaces intersecting with the boundary, see Figure 3.9. One approach to resolve this issue is the introduction of an additional boundary condition for boundaries $\partial\Omega$ where $\varphi\nabla\varphi \cdot \mathbf{n}_{\partial\Omega} > 0$. Thus,

$$\mathbf{u} = \mathbf{u}_\text{D} \quad \text{on } \partial\Omega|_{\varphi\nabla\varphi\cdot\mathbf{n}_{\partial\Omega}>0}, \tag{3.89}$$

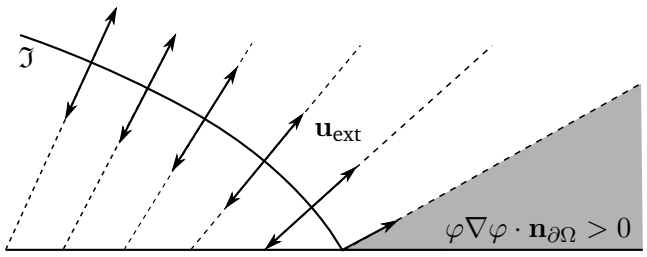


Figure 3.9.: The extension problem (3.88) is not well-posed in regions where $\varphi\nabla\varphi \cdot \mathbf{n}_{\partial\Omega} > 0$ (grey area).

where $\mathbf{u}_{\text{D}}$ in context of two-phase flow simulations is provided by the solution of the flow solver. Another approach is the extension to a vanishing-viscosity solution, see Equation (3.93).

The basic approaches to solve the extension problem are similar to the methods for reinitialization, see Section 3.6.2. They include direct methods (Chessa et al., 2002; Gibou and Fedkiw, 2005; Sauerland, 2013) and fast-marching (Adalsteinsson and Sethian, 1999) or fast-sweeping methods (Aslam et al., 2014). Another approach is to rewrite the PDE (3.88) as the steady-state solution of an advection equation (Chen et al., 1997b). The reformulation to an elliptic PDE is the basis of the method presented in the following.

Extension-Velocity Problem as an Elliptic PDE In Utz and Kummer (2018) the idea of reformulation the Eikonal equation (3.73) into a elliptic PDE is transferred to the extension problem (3.88). In Utz (2018) the problem statement is slightly modified by considering the normalized level-set gradient $\overline{\nabla\varphi} = \frac{\nabla\varphi}{|\nabla\varphi|}$ yielding

$$\overline{\nabla\varphi} \cdot \nabla u = 0 \quad \text{in } \Omega. \tag{3.90}$$

This problem results in more stable solutions, since it is better conditioned in regions where $|\nabla\varphi|$ is small. As for the elliptic reinitialization, the modified extension problem (3.90) is described as the minimization problem

$$E(\varphi) = \int_{\Omega} \psi(\overline{\nabla\varphi} \cdot \nabla u)\, \mathrm{d}V \to \min \quad \text{in } \Omega \tag{3.91a}$$

$$\text{s.t. } u = u_{\mathfrak{I}} \quad \text{on } \mathfrak{I}, \tag{3.91b}$$

where the potential is given by $\psi(s) = \frac{s^2}{2}$ with the solution $s = 0$. Again, the minimization problem (3.91) can be rewritten in a PDE

$$\nabla \cdot \left(\left(\overline{\nabla\varphi} \otimes \overline{\nabla\varphi} \right) \nabla u \right) = 0 \quad \text{in } \Omega \setminus \mathfrak{I} \tag{3.92a}$$

$$\left(\left(\overline{\nabla\varphi} \otimes \overline{\nabla\varphi} \right) \nabla u \right) \cdot \mathbf{n}_{\partial\Omega} = 0 \quad \text{on } \partial\Omega \tag{3.92b}$$

$$u = u_{\text{D}} \quad \text{on } \partial\Omega_{\text{D}} \tag{3.92c}$$

$$u = u_{\mathfrak{I}} \quad \text{on } \mathfrak{I}, \tag{3.92d}$$

where the Dirichlet boundary condition (3.89) is additionally incorporated. This boundary value problem describes the steady-state limit of an anisotropic diffusion problem, where $\mathbf{u}$ is

linear. Thus, the discretized problem can be solved by a suitable linear solver in contrast to the reinitialization. For the DG discretization using an upwinding flux and the choice of the penalty parameter the reader is referred to Utz (2018).

One should note that the reformulation into the PDE (3.92) requires that the original problem (3.88) is well-posed, i.e. the level-set fulfils the signed-distance property. However, due to interface evolution and reinitialization this property is not exactly given. In order to stabilize the above method Utz (2018) extends the problem (3.90) via the vanishing viscosity solution

$$\lim_{\epsilon \to 0} \overline{\nabla \varphi} \cdot \nabla u - \epsilon \Delta u = 0 \quad \text{in } \Omega. \tag{3.93}$$

The additional isotropic viscosity term ϵ is chosen to be $\epsilon \leq 10^{-2}$.

3.6.4. Interface Evolution Algorithm

This section provides an overview on the procedure performed to compute the new interface position at t^{n+1}, i.e. computing the level-set fields $\varphi^{\mathrm{dg},n+1}$ and $\varphi^{\mathrm{c0},n+1}$ from the the previous time step t^n. The whole procedure UpdateLevelSet($\mathbf{u}^n$, Δt) is given as pseudocode in Algorithm 2 with corresponding references to issues presented above. Note that the operations are restricted to the cells forming the narrow-band around the interface, see Figure 3.7.

Algorithm 2 Interface evolution algorithm

1: **procedure** UpdateLevelSet($\mathbf{u}^n$, Δt)
2: $\quad \mathbf{u}_{\rho\mathrm{Aver}} \leftarrow$ GetMeanVelocityFromXDGField($\mathbf{u}^n$) ▷ Apply (3.74)
3: $\quad \varphi^{\mathrm{dg},n+1} \leftarrow$ LevelSetEvolution($\varphi^{\mathrm{dg},n}, \mathbf{u}_{\rho\mathrm{Aver}}, \Delta t$)
4: $\quad \mathfrak{K}_{\mathrm{cc}}{}^n \leftarrow$ GetCutCells()
5: $\quad \mathfrak{K}_{\mathrm{near}}{}^n \leftarrow$ GetNeighbourCells(cc)
6: $\quad \varphi^{\mathrm{dg},n+1*} \leftarrow$ FastMarchReInit($\varphi^{\mathrm{dg},n+1}, \mathfrak{K}_{\mathrm{cc}}{}^n, \mathfrak{K}_{\mathrm{near}}{}^n$)
▷ Algorithm 1, optional for a predefined interval
7: $\quad \varphi^{\mathrm{c0},n+1} \leftarrow$ ContinuityProjection($\varphi^{\mathrm{dg},n+1}, \mathfrak{K}_{\mathrm{cc}}{}^n \cup \mathfrak{K}_{\mathrm{near}}{}^n$)
▷ Solve (3.75) on narrow-band
8: $\quad \varphi^{\mathrm{c0},n+1*} \leftarrow$ EnforceMassConservation($\varphi^{\mathrm{c0},n+1}, \varphi^{\mathrm{c0},n}$) ▷ Apply (3.94), optional
9: $\quad \mathfrak{A}^{n+1}, \mathfrak{B}^{n+1}, \mathfrak{I}^{n+1} \leftarrow$ UpdateDomains($\varphi^{\mathrm{c0},n+1}$)
10: $\quad$ UpdateNarrowBand()
11: $\quad$ **return** $\varphi^{\mathrm{c0},n+1}$
12: **end procedure**

The procedure LevelSetEvolution in line 3 is interchangeable according to the chosen level-set evolution algorithm, see Algorithm 3 and Algorithm 4. In UpdateLevelSet there are two post-processing operations that are optional: The reinitialization (Algorithm 1) in line 6, which may be applied on a predefined time step interval, and a global mass correction in line 8 applied every time step. If not stated otherwise, both operations are not performed.

Global mass conservation In general, level-set methods are not mass conserving during the advection step and may lead to an accumulating error in the mass conservation. In this work an optional mass correction step may be performed explicitly in order to regain mass conservation in a global sense. Following the idea of Smolianski (2001) the correction is implemented by adding a signed constant $c_{\varphi,\text{corr}}$ to the whole level-set field. For our implementation the constant is given by

$$c_{\varphi,\text{corr}} = \frac{\mathfrak{A}(0) - \mathfrak{A}(t^n)}{|\mathfrak{I}|\,(t^n)}, \tag{3.94}$$

where $|\mathfrak{I}|$ denotes the length of the interface. Note that a positive value, i.e. $\mathfrak{A}(0) > \mathfrak{A}(t^n)$, expands the interface and a negative value, i.e. $\mathfrak{A}(0) < \mathfrak{A}(t^n)$, shrinks it. According to Smolianski (2001) such a correction is only justified if the level-set field satisfies the signed-distance property in some vicinity of $\mathfrak{I}$ and $|c_{\varphi,\text{corr}}|$ is not greater than the interpolation error of the interface. Therefore, we apply the mass conservation after the reinitialization step and only if $|c_{\varphi,\text{corr}}|$ is smaller than $\mathcal{O}(h^{p+1})$.

Fast-Marching Level-Set Evolution In Algorithm 3 a fast-marching based level-set evolution procedure is presented that provides a signed-distance level-set on a narrow-band. The far field values are set either to -1 or $+1$ according to their corresponding domain $\mathfrak{A}$ or $\mathfrak{B}$. The basic notion of this algorithm is to construct the extension-velocity field $\mathbf{u}_{\text{ext}}$ in two steps: First, the cut-cells are computed monolithic by the elliptic PDE approach (3.92) and second, the construction onto the neighbouring cells is performed by a geometric fast-marching method. The advection equation (2.14) is discretized by an upwinding scheme in space and with a Runge-Kutta scheme (TVD3) in time. After that, the level-set is additionally stabilized by penalizing the jumps of the level-set at inner edges, i.e. $\int_{\Gamma_{\text{int}}} \left[\!\left[\varphi^{\text{dg}\,n+1}\right]\!\right] [\![\vartheta]\!] \,\mathrm{d}S$ with test function ϑ, via an implicit Euler scheme with $\Delta t_{pnlty} = 0.001\Delta t$. Thus, the deviation to the projected continuous level-set field φ^{c0} used for the spatial discretization is reduced.

3.6.5. Fourier-Based Interface Representation

This developed approach combines an explicit interface representation via markers and the analytical formulation of the interface provided by a discrete fast Fourier transformation (FFT). Thus, the normal and curvature fields may be computed exactly from the analytical representation and projected on DG fields. However, such a representation is limited to wave and circle-like shapes of the interface, see Figure 3.10. In order to obtain an analytical formulation of the interface in terms of a Fourier representation

$$F(x) = \sum_{m=0}^{N_s} \tilde{fc}_m \mathrm{e}^{\mathrm{i}mx} = \sum_{m=0}^{N_s} \tilde{fc}_m (\cos(mx) + \mathrm{i}\sin(mx)), \tag{3.95}$$

the interface is given by N_s equidistant sample points $\mathbf{q}_s$. Thus, the discrete FFT provides N_s Fourier coefficients fc_m for the corresponding modes $m = 0, ..., N_s - 1$. The level-set formulation for a wave-like interface, see Figure 3.10 on the left, is then given by

$$\varphi(x, y) = y - F_{\text{planar}}(x). \tag{3.96}$$

Algorithm 3 Fast-marching level-set evolution algorithm on a narrow-band around the interface

procedure FASTMARCHINGLEVELSETEVOLUTION($\varphi^{\mathrm{dg},n}$, $\mathbf{u}_{\rho\mathrm{Aver}}$, Δt)
  $\mathfrak{K}_{\mathrm{cc}}{}^n \leftarrow$ GetCutCells()
  $\mathfrak{K}_{\mathrm{near}}{}^n \leftarrow$ GetNeighbourCells($\mathfrak{K}_{\mathrm{cc}}{}^n$)
  $\mathfrak{K}_{\mathrm{far}}{}^n \leftarrow \mathfrak{K}_h \setminus (\mathfrak{K}_{\mathrm{cc}}{}^n \cup \mathfrak{K}_{\mathrm{near}}{}^n)$ ▷ $\mathfrak{K}_{\mathrm{cc}}{}^n \cup \mathfrak{K}_{\mathrm{near}}{}^n$: narrow-band
  $\mathfrak{K}^n_{\mathrm{far}\mathfrak{A}} = -1; \mathfrak{K}^n_{\mathrm{far}\mathfrak{B}} = +1$
  $\nabla\varphi^{\mathrm{dg},n} \leftarrow$ ComputeGradient($\mathfrak{K}_{\mathrm{near}}{}^n$)
  if $\left\|\nabla\varphi^{\mathrm{dg},n}\right\| < 10^{-12}$ on $\mathfrak{K}_{\mathrm{near}}{}^n$ **then**
    $\varphi^{\mathrm{dg}*} \leftarrow$ FastMarchReInit($\mathfrak{K}_{\mathrm{near}}{}^n$) ▷ Algorithm 1
  end if
  $\nabla\varphi^{\mathrm{dg},*} \leftarrow$ ComputeGradient($\mathfrak{K}_{\mathrm{near}}{}^n$)
  $\mathbf{u}_{\mathrm{ext}}{}^* \leftarrow$ EllipticExtensionVelocity($\mathfrak{K}_{\mathrm{cc}}{}^n$) ▷ solve PDE (3.92)
  $\mathbf{u}_{\mathrm{ext}} \leftarrow$ FastMarchingExtensionVelocity($\mathfrak{K}_{\mathrm{near}}{}^n$) ▷ by a geometric approach
  $\varphi^{\mathrm{dg},n+1} \leftarrow$ Advect($\mathbf{u}_{\mathrm{ext}}, \Delta t, \mathfrak{K}_{\mathrm{cc}}{}^n \cup \mathfrak{K}_{\mathrm{near}}{}^n$)
    ▷ (2.14) with upwind stabilization, Runge-Kutta method TVD3
  $\varphi^{\mathrm{dg},n+1*} \leftarrow$ JumpPenalization($\varphi^{\mathrm{dg},n+1}$) ▷ via implicit Euler
  return $\varphi^{\mathrm{dg},n+1*}$
end procedure

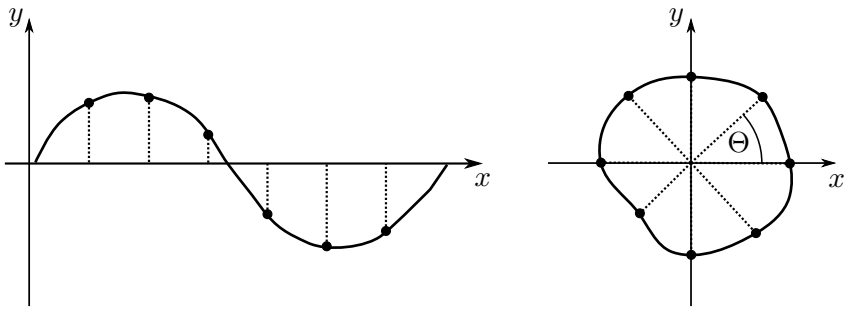


Figure 3.10.: Fourier-series representation for a wave-like (left) and a circle-shaped (right) interface.

Considering a circle-like interface, see Figure 3.10 on the right, the level-set is given in terms of polar coordinates $r = r(x, y)$ and $\theta = \theta(x, y)$ w.r.t its geometric centre by

$$\varphi(x, y) = r - F_{\mathrm{polar}}(\theta). \tag{3.97}$$

Both formulations allow a direct projection on a DG field and furthermore analytic expressions for the normal field $\mathbf{n}_\mathfrak{I}$ (3.22) and curvature field κ (3.23). Since it is given as a Fourier series, the minimal resolved wavelength of the interface may be adjusted according to the projected Fourier modes m. The interface evolution is then performed by advecting the material sample points $\mathbf{q}_s$ with the density-averaged velocity field $\mathbf{u}_{\rho\mathrm{Aver}}$. The entire Fourier-based interface evolution is given in Algorithm 4. Note that after the explicit advection the sample points $\mathbf{q}_s$ are no longer equally spaced not allowing a standard discrete FFT. So the sample points need to be rearranged equidistantly on the periodic domain.

Algorithm 4 Fourier-based level-set evolution algorithm

procedure FOURIERLEVELSETEVOLUTION($\varphi^{\mathrm{dg},n}$, $\mathbf{u}_{\rho\mathrm{Aver}}$, Δt)
  internal $\mathbf{q}_s^n$, fc_m^n
  $\mathbf{u}_s^n \leftarrow$ EvaluateVelocityAtSamplePoints($\mathbf{u}_{\rho\mathrm{Aver}}, \mathbf{q}_s^n$)
  $\mathbf{q}_s^* \leftarrow$ Advect($\mathbf{q}_s^n, \mathbf{u}_s^n$) ▷ Runge-Kutta scheme
  $\mathbf{q}_s^{n+1} \leftarrow$ InterpolateOntoFourierPoints($\mathbf{q}_s^*$) ▷ cubic spline interpolation
  $fc_m^{n+1} \leftarrow$ discreteFFT($\mathbf{q}_s^{n+1}$) ▷ Eq. (3.95)
  $\mathbf{q}_s^{n+1*} \leftarrow$ SmoothSamplePoints($\mathbf{q}_s^{n+1}, fc_m^{n+1}, m_{max}$)
    ▷ m_{max} corresponds to the minimal resolved wavelength
  $\varphi^{\mathrm{dg},n+1} \leftarrow$ ProjectToDGLevelSet(fc_m^{n+1}, m_{max}) ▷ Eq. (3.96) or Eq. (3.97)
  return $\varphi^{\mathrm{dg},n+1}$
end procedure

3.7. Discretization of the Surface Tension Force

In this section the various discretization options of the surface tension force that are used in this work are presented. Recalling the discretization of the right-hand side of the Navier-Stokes equation (3.48) the surface tension force is given by

$$\oint_{\mathfrak{I}} \sigma\kappa\mathbf{n}_{\mathfrak{I}} \cdot \mathbf{v} \, \mathrm{d}S. \tag{3.98}$$

In this formulation one needs to compute the curvature field κ from the level-set field φ via Bonnet's formula (3.23). Note that this includes the numerical evaluation of second order derivatives of φ. In the context of a singed-distance level-set method such an evaluation is highly sensitive to minor inaccuracies in the level-set field as shown in Kummer and Warburton (2016). An example is provided, where the signed-distance level-set function for a circle is projected onto the broken polynomial DG space. This projection alone introduces an L^∞-error of around 10% in the curvature. Note that the projection of the quadratic circle formulation is exact up to round-off errors. Therefore, Kummer and Warburton (2016) developed a patch-recovery filter to reduce the errors in the curvature evaluation, which is presented in Section 3.7.2.

However, it is not necessary to evaluate the curvature, as briefly discussed in Section 3.4.2. In this case the surface tension force (3.98) is rewritten via the Laplace-Beltrami operator that for a sufficient smooth function f is defined by

$$\Delta_{\mathfrak{I}} f := \nabla_{\mathfrak{I}} \cdot \nabla_{\mathfrak{I}} f. \tag{3.99}$$

Now let $\mathrm{id}_{\mathfrak{I}}$ be the identity mapping on $\mathfrak{I}$ with $\mathrm{id}_{\mathfrak{I}}(\mathbf{x}) = \mathbf{x}$ for $\mathbf{x} \in \mathfrak{I}$. According to differential geometry, see e.g. Gallot et al. (2004), it follows from $\nabla \mathrm{id}_{\mathfrak{I}} = \mathbf{I}$ and $\nabla_{\mathfrak{I}} := \mathbf{P}_{\mathfrak{I}} \nabla$ that

$$\kappa\mathbf{n}_{\mathfrak{I}} = \Delta_{\mathfrak{I}} \mathrm{id}_{\mathfrak{I}}. \tag{3.100}$$

Thus, the discretization in Equation (3.98) for a closed interface $\mathfrak{I}$ with (3.99) and (3.100) results after partial integration in

$$\oint_{\mathfrak{I}} \sigma \Delta_{\mathfrak{I}} \mathrm{id}_{\mathfrak{I}} \cdot \mathbf{v} \, \mathrm{d}S = -\oint_{\mathfrak{I}} \sigma \nabla_{\mathfrak{I}} \mathrm{id}_{\mathfrak{I}} : \nabla_{\mathfrak{I}} \mathbf{v} \, \mathrm{d}S. \tag{3.101}$$

The Laplace-Beltrami formulation ensures force conservation along closed interfaces via the boundary integral terms.

Semi-Implicit Discretization The formulation (3.101) is the basis for the semi-implicit treatment of the surface tension force introduced by Dziuk (1990). In the context of a level-set method this discretization is e.g. found in Gross and Reusken (2007) and Hysing (2006). For the derivation we follow the work of Hysing (2006). There, an explicit time integration for the surface tension force $\mathbf{f}_{\text{st}}$ is employed in which the spatial integration is performed on the previous time step t^n with

$$\mathbf{f}_{\text{st}} = \oint_{\mathfrak{I}^n} \sigma\kappa\mathbf{n}_{\mathfrak{I}}{}^n \cdot \mathbf{v} \, \mathrm{d}S = \oint_{\mathfrak{I}^n} \sigma\nabla_{\mathfrak{I}}\mathrm{id}_{\mathfrak{I}}^n : \nabla_{\mathfrak{I}}\mathbf{v} \, \mathrm{d}S. \tag{3.102}$$

The term is added to the right-hand side as a source term. Following the work of Bänsch (2001) the new interface position $\mathrm{id}_{\mathfrak{I}}^{n+1}$ is rewritten as an expression of the old position $\mathrm{id}_{\mathfrak{I}}^n$ with

$$\mathrm{id}_{\Gamma}^{n+1} = \mathrm{id}_{\Gamma}^n + \Delta t \mathbf{u}^{n+1}. \tag{3.103}$$

Introducing expression (3.103) into the explicit discretization (3.102) yields a semi-implicit discretization given by

$$\mathbf{f}_{\text{st}} = -\oint_{\mathfrak{I}^n} \sigma\nabla_{\mathfrak{I}}\mathrm{id}_{\mathfrak{I}}^n : \nabla_{\mathfrak{I}}\mathbf{v} \, \mathrm{d}S - \Delta t \oint_{\mathfrak{I}^n} \sigma\nabla_{\mathfrak{I}}\mathbf{u}^{n+1} : \nabla_{\mathfrak{I}}\mathbf{v} \, \mathrm{d}S. \tag{3.104}$$

The additional term including the surface gradient of $\mathbf{u}^{n+1}$ can be assembled as a positive definite contribution to the iteration (operator) matrix of the system. This term represents a diffusion operator working in the tangential direction of $\mathfrak{I}$ and gives a clear advantage over a pure explicit discretization. This results in a more physical implementation of capillary effects since an increased coefficient of surface tension generates more interface diffusion, that is a stiffer system, instead of a larger destabilizing source term (Hysing, 2006).

One should note that Hysing (2006) combines the semi-implicit discretization with a regularization of the surface tension force term via the continuum surface force (CSF), see (Brackbill et al., 1992). In this case the surface tension force is rewritten as a volumetric force in Ω with

$$\mathbf{f}_{\text{st}} = \oint_{\mathfrak{I}} \sigma\kappa\mathbf{n}_{\mathfrak{I}} \cdot \mathbf{v} \, \mathrm{d}S = \oint_{\Omega} \sigma\kappa\mathbf{n}_{\mathfrak{I}} \cdot \mathbf{v}\delta(\mathfrak{I}, \mathbf{x}) \, \mathrm{d}S, \tag{3.105}$$

where the Dirac delta function $\delta(\mathfrak{I}, \mathbf{x})$ is regularized by a continuous delta function δ_ϵ with support ϵ. In this work the CSF approach is not followed, since the XDG method provides high-order numerical integration techniques in order to accurately evaluate the desired interface integrals, see Section 3.3.2.

A Note on the Time Integration Considering the time integration of the surface tension force within the coupled solver structure, see Section 3.9, the evolution of the interface is performed prior to the construction of the linearized system. Thus, the spatial discretization (3.38) is done for the new interface position $\mathfrak{I}^{n+1}$. Note that the new position may be either explicitly or implicitly coupled with the solution of the Navier-Stokes equations. This is discussed in Section 3.9, where the overall solver structure is presented.

3.7.1. Implementation of a Dissipative Interface Model

As noted before the semi-implicit discretization provides an additional interface diffusion term. However, the dissipative effect only scales with the time step size and does not take other properties of the interface into account. Therefore, we are considering the discretization of a Boussinesq-Scriven interface model, which introduces thermodynamic consistently dissipative terms into the surface stress tensor providing a more general approach in comparison to the semi-implicit formulation.

Recalling the definition of the surface tension force on the right-hand side of the momentum jump condition (2.17), this can be rewritten in the following surface divergence form with $\sigma = \text{const}$

$$\sigma\kappa\mathbf{n}_\mathfrak{I} = \nabla_\mathfrak{I} \cdot (\sigma\mathbf{P}_\mathfrak{I}). \tag{3.106}$$

The tensor $\sigma\mathbf{P}_\mathfrak{I}$ can be interpreted as the isotropic part of a general surface stress tensor $\mathbf{S}_\mathfrak{I}$ with

$$\mathbf{S}_\mathfrak{I} = \sigma\mathbf{P}_\mathfrak{I} + \mathbf{S}_\mathfrak{I}^D, \tag{3.107}$$

which is described similar to the Cauchy stress tensor for the bulk phases with $\mathbf{S} = -p\mathbf{I} + \mathbf{S}^D$. The most general linear relation between the surface stress tensor and the surface rate of deformation tensor that is consistent with the principle of frame indifference is (Slattery et al., 2007)

$$\mathbf{S}_\mathfrak{I} = \left(\alpha + \tilde{\lambda}_\mathfrak{I}\left(\nabla_\mathfrak{I} \cdot \mathbf{u}_\mathfrak{I}\right)\right)\mathbf{P}_\mathfrak{I} + 2\mu_\mathfrak{I}\mathbf{D}_\mathfrak{I}(\mathbf{u}_\mathfrak{I}), \tag{3.108}$$

where $\mathbf{D}_\mathfrak{I}(\mathbf{u}_\mathfrak{I}) = \frac{1}{2}\mathbf{P}_\mathfrak{I}\left(\nabla_\mathfrak{I}\mathbf{u}_\mathfrak{I} + \nabla_\mathfrak{I}\mathbf{u}_\mathfrak{I}{}^\mathrm{T}\right)\mathbf{P}_\mathfrak{I}$ is the surface rate of deformation tensor. The jump entropy inequality for a dividing surface implies

$$\alpha = \sigma, \quad \lambda_\mathfrak{I} \equiv \tilde{\lambda}_\mathfrak{I} + \mu_\mathfrak{I} > 0, \quad \mu_\mathfrak{I} > 0. \tag{3.109}$$

Thus, the interface stress tensor takes the form of the linear Boussinesq-Scriven model

$$\mathbf{S}_\mathfrak{I}^{\mathrm{BS}} = \sigma\mathbf{P}_\mathfrak{I} + \tilde{\lambda}_\mathfrak{I}\left(\nabla_\mathfrak{I} \cdot \mathbf{u}_\mathfrak{I}\right)\mathbf{P}_\mathfrak{I} + 2\mu_\mathfrak{I}\mathbf{D}_\mathfrak{I}(\mathbf{u}_\mathfrak{I}) \tag{3.110}$$

with $\tilde{\lambda}_\mathfrak{I} = \lambda_\mathfrak{I} - \mu_\mathfrak{I}$, where $\lambda_\mathfrak{I}$ and $\mu_\mathfrak{I}$ are referred to as the interface dilatational viscosity and interface shear viscosity, respectively.

For the Boussinesq-Scriven interface model (3.111) we propose the following discretization of the surface tension force $\nabla_\mathfrak{I} \cdot \mathbf{S}_\mathfrak{I}^{\mathrm{BS}}$: Multiplying (3.111) by a suitable test function $\mathbf{v}$ and integration by parts over the interface $\mathfrak{I}$ yields

$$\begin{aligned}
\oint_\mathfrak{I} \nabla_\mathfrak{I} \cdot \mathbf{S}_\mathfrak{I}^{\mathrm{BS}} \cdot \mathbf{v}\, \mathrm{d}S = &- \oint_\mathfrak{I} \sigma\mathbf{P}_\mathfrak{I} : \nabla_\mathfrak{I}\mathbf{v}\, \mathrm{d}S + \int_{\mathfrak{I}\cap\Gamma_{\text{int}}} \sigma\left\{\{\boldsymbol{\tau}\}\right\} \cdot [\![\mathbf{v}]\!]\, \mathrm{d}l \\
&- \oint_\mathfrak{I} \tilde{\lambda}_\mathfrak{I}\left(\nabla_\mathfrak{I} \cdot \mathbf{u}\right)\mathbf{P}_\mathfrak{I} : \nabla_\mathfrak{I}\mathbf{v}\, \mathrm{d}S + \int_{\mathfrak{I}\cap\Gamma_{\text{int}}} \left\{\left\{\tilde{\lambda}_\mathfrak{I}\left(\nabla_\mathfrak{I} \cdot \mathbf{u}\right)\boldsymbol{\tau}\right\}\right\} \cdot [\![\mathbf{v}]\!]\, \mathrm{d}l \\
&- \oint_\mathfrak{I} \mu_\mathfrak{I}\left(\nabla_\mathfrak{I}\mathbf{u} + \nabla_\mathfrak{I}\mathbf{u}^\mathrm{T}\right)\mathbf{P}_\mathfrak{I} : \nabla_\mathfrak{I}\mathbf{v}\, \mathrm{d}S \\
&+ \int_{\mathfrak{I}\cap\Gamma_{\text{int}}} \left\{\left\{\mu_\mathfrak{I}\left(\nabla_\mathfrak{I}\mathbf{u} + \nabla_\mathfrak{I}\mathbf{u}^\mathrm{T}\right)\boldsymbol{\tau}\right\}\right\} \cdot [\![\mathbf{v}]\!]\, \mathrm{d}l
\end{aligned} \tag{3.111}$$

Note that in the first line the isotropic part (3.50) is regained. Comparing the first term of the third line in (3.111), i.e. $\oint_\mathfrak{I} \mu_\mathfrak{I} \nabla_\mathfrak{I} \mathbf{u} \mathbf{P}_\mathfrak{I} : \nabla_\mathfrak{I} \mathbf{v} \, \mathrm{d}S$, with the semi-implicit discretization (3.104), we note that the identity term of $\mathbf{P}_\mathfrak{I} = (\mathbf{I} - \mathbf{n}_\mathfrak{I} \otimes \mathbf{n}_\mathfrak{I})$ is equal to the additional semi-implicit term scaled by different factors.

3.7.2. L^2-Based Patch Recovery Filter

In this section the patch recovery method based on an L^2-projection and developed by Kummer and Warburton (2016) is presented. The original patch recovery by Zienkiewicz and Zhu (1995) used a nodal projection in order to regain super-convergence for stresses and gradient-values in post-processing. In Kummer and Warburton (2016) the patch recovery is used for filtering the the level-set field φ and its derivatives. The patch recovery operator is in general defined as an L^2-projection from a polynomial space of degree q onto a polynomial space of degree r with $r \geq q$:

$$\mathrm{prc}_w : \mathbb{P}_q({\mathfrak{K}_h}^w) \to \mathbb{P}_r({\mathfrak{K}_h}^w). \tag{3.112}$$

Here, w denotes the width of the patch where the operator is defined. In the context of level-set filtering the patch either only describes the set of cut-cells $\mathfrak{K}_{\mathrm{cc}}$, i.e. width $w = 0$, or it describes both the cut-cells and the neighbouring cells $\mathfrak{K}_{\mathrm{near}}$, i.e. $w = 1$. For a cell $K \in {\mathfrak{K}_h}^w$ and a polynomial function $u \in \mathbb{P}_q({\mathfrak{K}_h}^w)$ the patch recovery operation $\mathrm{prc}_w(u) =: v \in \mathbb{P}_r({\mathfrak{K}_h}^w)$ is then defined by

$$v|_K := \begin{cases} \mathrm{Proj}_{\mathbb{P}_r(\{Q_K\})}(u|_{Q_K})\Big|_K & \text{if } K \in {\mathfrak{K}_h}^w, \\ 0 & \text{if } K \notin {\mathfrak{K}_h}^w. \end{cases} \tag{3.113}$$

The L^2-projection $\mathrm{Proj}_{\mathbb{P}_r(\{Q_K\})}(u)$ describes the projection onto a composite cell Q_K that denotes the union of the cell K and its neighbours in ${\mathfrak{K}_h}^w$. Thus, $\mathrm{Proj}_{\mathbb{P}_r(\{Q_K\})}(u|_{Q_K})$ performs a projection of a broken polynomial function on K and its neighbours, i.e. $u|_{Q_K}$, onto a continuous polynomial on Q_K, i.e. $\mathbb{P}_r(\{Q_K\})$. The result in cell K, denoted by $v|_K$, is then regained by a restriction of the polynomial on Q_K to the cell K. This operation is done for every cell K in ${\mathfrak{K}_h}^w$. Furthermore, the whole patch recovery operation on ${\mathfrak{K}_h}^w$ is performed in multiple sweeps. Details on the implementation of the projection operator and the configurable options are given in Kummer and Warburton (2016).

Filter Configuration for Curvature Computation Filtering the numerical computation of the curvature field κ, Bonnet's formula (3.23) can be expressed as

$$\nabla \cdot \left(\frac{\nabla \varphi}{\|\nabla \varphi\|} \right) = \frac{\mathrm{tr}(\partial^2 \varphi)}{\|\nabla \varphi\|} - \frac{\nabla \varphi^{\mathsf{T}} (\partial^2 \varphi) \nabla \varphi}{\|\nabla \varphi\|^3}, \tag{3.114}$$

where $\partial^2 \varphi$ denotes the level-set Hessian. According to the results of Kummer and Warburton (2016) the optimal filter configuration for κ is given as follows. The patch width is set to zero and the projection is performed with the same polynomial degree, i.e. $r = q$. Both the level-set gradient $\nabla \varphi$ and the Hessian are filtered. The algorithm is performed with three sweeps.

3.7.3. Capillary Time Step Restriction

The numerical simulation of two-phase flows with a dividing interface imposed by a surface tension σ is inherently restricted by the time step size due to capillary waves propagating on the interface. The phase velocity of such waves for the inviscid case is given by

$$c_\sigma = \frac{\omega_0}{k} = \sqrt{\frac{\sigma k}{\rho_\mathfrak{A} + \rho_\mathfrak{B}}},$$

where ω_0 denotes the natural frequency and k the corresponding wave number. Based on this velocity Brackbill et al. (1992) stated the following time step restriction $\Delta t_\sigma^{\text{Brackbill}}$ in order to temporally resolve the shortest spatially resolved wavelength with $\lambda_{\text{min}} = 2h$:

$$\Delta t_\sigma^{\text{Brackbill}} = \frac{h}{2c_\sigma} = \sqrt{\frac{(\rho_\mathfrak{A} + \rho_\mathfrak{B})h^3}{4\pi\sigma}}. \tag{3.115}$$

However, a revised capillary time step constraint by Denner and Wachem (2015) based on different analysis viewpoints such as the principle of domain dependence (Lewy et al., 1928), the Nyquist-Shannon theorem (Marks, 1991), and opposing travelling waves reads

$$\Delta t_\sigma^{\text{stat}} = \frac{h}{\hat{c}_\sigma} = \sqrt{\frac{(\rho_\mathfrak{A} + \rho_\mathfrak{B})h^3}{2\pi\sigma}}, \quad \text{with } \hat{c}_\sigma = \sqrt{2}c_\sigma, \tag{3.116}$$

which is less restrictive than Equation (3.115). One should note that both restrictions are only applicable to the static case. Considering the dynamic case with an additional fluid motion parallel to the interface $\mathbf{u}_{\mathfrak{I},\parallel}$ Denner and Wachem (2015) proposes the following restriction

$$\Delta t_\sigma^{\text{dyn}} = \frac{h}{\hat{c}_\sigma + \left|\mathbf{u}_{\mathfrak{I},\parallel}\right|}. \tag{3.117}$$

Furthermore, Equation (3.117) may be generalized for arbitrary meshes according to

$$\Delta t_\sigma^{\text{dyn}} = \frac{\min|\mathbf{s}_\mathfrak{I}|}{\hat{c}_\sigma + \left|\mathbf{u}_{\mathfrak{I},\parallel}\right|}, \tag{3.118}$$

where $\mathbf{s}_\mathfrak{I}$ denotes a vector connecting neighbouring computational nodes. For our numerical simulations we restrict the time step size according to the revised static restriction, since in general we do not a priori know the developing tangential velocities at the interface. Thus, we extend the restriction (3.116) with the reasoning of Equation (3.118). Since approximating the interface, resp. the level-set field, by a polynomial basis of degree k we set the minimal resolved length to $\frac{h}{k+1}$. The resulting capillary time step restriction is given by

$$\Delta t_\sigma = \sqrt{\frac{(\rho_\mathfrak{A} + \rho_\mathfrak{B})\frac{h}{k+1}^3}{2\pi\sigma}}. \tag{3.119}$$

3.8. Discrete Energy Estimations for Transient Two-Phase Flow Problems

In this section an energy estimation for both the transient two-phase flow problem and the extended contact line problem are presented and the stability in the resulting energy norms is discussed. We follow the work of Reusken et al. (2017) to derive the energy estimates. For the sole two-phase setting we define the energy in the computational domain Ω as the sum of the kinetic bulk energy and surface energy with

$$E_{\text{sys}}(t) = \frac{1}{2}\int_{\Omega} \rho \left\|\mathbf{u}\right\|^2 \, \mathrm{d}V + \sigma|\mathfrak{I}(t)|. \tag{3.120}$$

The norm $\|\cdot\|^2$ denotes the Euclidean vector norm and $|\mathfrak{I}(t)|$ the area of the interface $\mathfrak{I}(t)$. We consider all boundaries to be imposed by the no-slip boundary condition. Note that we do not take any gravity forces into account in this section. Considering the change rate of the kinetic energy term we get

$$\begin{aligned}
\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega} \rho \left\|\mathbf{u}\right\|^2 \, \mathrm{d}V &= \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathfrak{A}(t)\dot{\cup}\mathfrak{B}(t)} \rho \left\|\mathbf{u}\right\|^2 \, \mathrm{d}V \\
&= \frac{1}{2}\int_{\mathfrak{A}(t)\dot{\cup}\mathfrak{B}(t)} \rho\frac{\partial}{\partial t}(\mathbf{u}\cdot\mathbf{u}) + \rho\mathbf{u}\cdot\nabla(\mathbf{u}\cdot\mathbf{u}) \, \mathrm{d}V \\
&= \int_{\mathfrak{A}(t)\dot{\cup}\mathfrak{B}(t)} \rho\frac{\partial \mathbf{u}}{\partial t}\cdot\mathbf{u} + \rho(\mathbf{u}\cdot\nabla\mathbf{u})\cdot\mathbf{u} \, \mathrm{d}V \\
&= \int_{\mathfrak{A}(t)\dot{\cup}\mathfrak{B}(t)} \rho\frac{\partial \mathbf{u}}{\partial t}\cdot\mathbf{u} + \rho\left(\nabla\cdot(\mathbf{u}\otimes\mathbf{u})\right)\cdot\mathbf{u} \, \mathrm{d}V,
\end{aligned} \tag{3.121}$$

where in the last line we used $\nabla\cdot\mathbf{u} = 0$ for incompressible flows. We can rewrite the last line in terms of the discretized Navier-Stokes equations (3.38). Therefore, we need to choose $\mathbf{u}^{n+1}$ as the test function $\mathbf{v}$ in (3.38) leading directly to $b(\vec{u}, p) = 0$. Thus, equation (3.121) may be written as

$$\begin{aligned}
&\int_{\mathfrak{A}(t)\dot{\cup}\mathfrak{B}(t)} \rho\frac{\partial \mathbf{u}}{\partial t}\cdot\mathbf{u} + \rho\left(\nabla\cdot(\mathbf{u}\otimes\mathbf{u})\right)\cdot\mathbf{u} \, \mathrm{d}V \\
&= m(\partial_t\mathbf{u}, \mathbf{u}) + t(\mathbf{u}, \mathbf{u}, \mathbf{u}) = -a(\mathbf{u}, \mathbf{u}) - \oint_{\mathfrak{I}} \sigma\mathbf{P}_{\mathfrak{I}} : \nabla_{\mathfrak{I}}\mathbf{u} \, \mathrm{d}S,
\end{aligned} \tag{3.122}$$

where we dropped the superscript $n+1$ and used the Laplace-Beltrami formulation (3.50) for the surface tension force. The viscous bilinear form reduces to

$$a(\mathbf{u}, \mathbf{u}) = -\int_{\Omega} \mu\left(\nabla_h\mathbf{u} : \nabla_h\mathbf{v} + \nabla_h\mathbf{u}^{\mathrm{T}} : \nabla_h\mathbf{v}\right) \, \mathrm{d}V = -\frac{1}{2}\int_{\Omega} \mu\mathrm{tr}(\mathbf{D}(\mathbf{u})^2) \, \mathrm{d}V \tag{3.123}$$

and the the surface tension force may be rewritten in terms of the interface area with

$$\oint_{\mathfrak{I}} \sigma\mathbf{P}_{\mathfrak{I}} : \nabla_{\mathfrak{I}}\mathbf{u} \, \mathrm{d}S = \oint_{\mathfrak{I}} \sigma\nabla_{\mathfrak{I}}\cdot\mathbf{u} \, \mathrm{d}S = \sigma\frac{\mathrm{d}}{\mathrm{d}t}|\mathfrak{I}(t)|. \tag{3.124}$$

Thus, for the energy in the whole computational domain (3.120) the following holds

$$\frac{\mathrm{d}}{\mathrm{d}t}E(t) = \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega} \rho \left\|\mathbf{u}\right\|^2 \, \mathrm{d}V + \frac{\mathrm{d}}{\mathrm{d}t}\sigma|\mathfrak{I}(t)| = -\frac{1}{2}\int_{\Omega} \mu\mathrm{tr}(\mathbf{D}(\vec{u})^2) \, \mathrm{d}V. \tag{3.125}$$

It is evident that the only dissipation results from the viscous terms in the bulk phases and that the term on the right-hand side is always negative. However, one should note that the energy estimate (3.125) only holds if the term regarding the change rate of the interface area $\frac{\mathrm{d}}{\mathrm{d}t}\sigma|\mathfrak{I}(t)|$ identically cancels out by Equation (3.124). This is only the case for a fully implicit discretization of the surface tension force. However, even for explicit or semi-implicit discretizations the energy estimate may hold if the time step size is sufficiently small, see e.g. Gerbeau and Lelièvre (2009).

One should note that there is no interface dissipation in the energy estimate (3.125). Considering the Boussinesq-Scriven model (3.111) the energy estimate extends to

$$\frac{\mathrm{d}}{\mathrm{d}t}E(t) = -\frac{1}{2}\int_\Omega \mu \mathrm{tr}(\mathbf{D}(\vec{u})^2)\ \mathrm{d}V - \lambda_\mathfrak{I}\int_\mathfrak{I}\mathrm{tr}(\nabla_\mathfrak{I}\vec{u})^2\ \mathrm{d}S - \frac{1}{2}\mu_\mathfrak{I}\int_\mathfrak{I}\mathrm{tr}(\mathbf{D}_\mathfrak{I}(\vec{u})^2)\ \mathrm{d}S, \tag{3.126}$$

where two additional viscosity terms at the interface are introduced, which corresponds to the interface dilatational viscosity $\lambda_\mathfrak{I}$ and interface shear viscosity $\mu_\mathfrak{I}$.

Extension to Contact Line Problems Extending the above energy estimate to two-phase flow problems exhibiting contact lines, i.e. $\partial\Omega$ includes a slip wall $\partial\Omega_\mathrm{S}$, the energy in the whole domain reads

$$E_\mathrm{sys}(t) = \frac{1}{2}\int_\Omega \rho\left\|\mathbf{u}\right\|^2\ \mathrm{d}V + \sigma|\mathfrak{I}(t)| + \gamma_\mathfrak{A}|\mathfrak{I}_\mathfrak{A}(t)| + \gamma_\mathfrak{B}|\mathfrak{I}_\mathfrak{B}(t)|, \tag{3.127}$$

where $\mathfrak{I}_\mathfrak{s}(t) = \partial\Omega_\mathfrak{s}\cup\partial\Omega_\mathrm{S}$ denotes the wetted area of species $\mathfrak{s}$ and $\gamma_\mathfrak{s}$ the corresponding surface tension coefficient, see Section 2.4. The resulting discrete kinetic energy change rate is given by

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\int_\Omega \rho\left\|\mathbf{u}\right\|^2\ \mathrm{d}V = -\tilde{a}(\mathbf{u},\mathbf{u}) - \oint_\mathfrak{I}\sigma\mathbf{P}_\mathfrak{I} : \nabla_\mathfrak{I}\mathbf{u}\ \mathrm{d}S - \int_L \sigma\cos(\theta_\mathrm{stat})\mathbf{u}\cdot\mathbf{n}_L\ \mathrm{d}l. \tag{3.128}$$

The extended viscous bilinear form $\tilde{a}(-,-)$ (3.53) including the slip boundary $\partial\Omega_\mathrm{S}$ reduces for $\mathbf{v}=\mathbf{u}$ and $(\mathbf{u}\cdot\mathbf{n}_\mathrm{S}) = 0$ to

$$\tilde{a}(\mathbf{u},\mathbf{u}) = -\frac{1}{2}\int_\Omega \mu\mathrm{tr}(\mathbf{D}(\mathbf{u})^2)\ \mathrm{d}V - \int_{\Gamma_\mathrm{S}}\beta_S\left\|\mathbf{P}_\mathrm{S}\mathbf{u}\right\|^2\ \mathrm{d}S - \int_L \beta_L(\mathbf{u}\cdot\mathbf{n}_L)^2\ \mathrm{d}l. \tag{3.129}$$

The surface tension term regarding Young's equation at the contact line L in Equation (3.128) is rewritten using the Stokes theorem in the plane containing the slip wall $\partial\Omega_\mathrm{S}$, where ∇_S denotes the divergence operator in this plane (Reusken et al., 2017). Thus, with $\mathfrak{I}_\mathfrak{A}\subset\partial\Omega_\mathrm{S}$ and $L = \partial\mathfrak{I}_\mathfrak{A}$ we get

$$\begin{aligned}
&\sigma\cos(\theta_\mathrm{stat})\int_L \mathbf{u}\cdot\mathbf{n}_L\ \mathrm{d}l = \sigma\cos(\theta_\mathrm{stat})\int_{\mathfrak{I}_\mathfrak{A}}\nabla_\mathrm{S}\cdot\mathbf{u}\ \mathrm{d}S\\
&= \sigma\cos(\theta_\mathrm{stat})\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathfrak{I}_\mathfrak{A}} 1\ \mathrm{d}S = \sigma\cos(\theta_\mathrm{stat})\frac{\mathrm{d}}{\mathrm{d}t}|\mathfrak{I}_\mathfrak{A}(t)|\\
&= -\frac{\mathrm{d}}{\mathrm{d}t}\left(\gamma_\mathfrak{A}|\mathfrak{I}_\mathfrak{A}(t)| + \gamma_\mathfrak{B}|\mathfrak{I}_\mathfrak{B}(t)|\right).
\end{aligned} \tag{3.130}$$

We applied Young's relation (2.22) and used that $\frac{\mathrm{d}}{\mathrm{d}t}\left(|\mathfrak{I}_{\mathfrak{A}}(t)| + |\mathfrak{I}_{\mathfrak{B}}(t)|\right) = \frac{\mathrm{d}}{\mathrm{d}t}\left(|\partial\Omega_{\mathrm{S}}|\right) = 0$ to derive the last line of Equation (3.130). Combining equations (3.127) to (3.130) results in the following energy estimate for two-phase flow problems with dynamic contact lines incorporated via the GNBC:

$$\frac{\mathrm{d}}{\mathrm{d}t}E(t) = -\frac{1}{2}\int_{\Omega}\mu\mathrm{tr}(\mathbf{D}(\vec{u})^2)\,\mathrm{d}V - \int_{\Gamma_{\mathrm{S}}}\beta_S\left\|\mathbf{P}_{\mathrm{S}}\mathbf{u}\right\|^2\,\mathrm{d}S - \int_{L}\beta_L(\mathbf{u}\cdot\mathbf{n}_L)^2\,\mathrm{d}l. \tag{3.131}$$

Again we note that the surface energies in (3.127) may not vanish identically for arbitrary discretizations of the surface tension force.

3.9. The XNSE-Solver Structure

In this section the overall structure of the developed extended-Navier-Stokes-Equations-solver (XNSE-solver) is presented and further details are discussed on solving the linearized system (see Section 3.9.1) and adaptively refining the mesh (see Section 3.9.2).

Algorithm 5 One time step of the XNSE-solver

procedure RunSolverOneStep(n,t^n,Δt) ▷ solve φ^{c0}, $\mathbf{u}$ and p for $t^{n+1} = t^n + \Delta t$
  $\varphi^{\mathrm{c0},n}, \mathbf{u}^n, p^n \leftarrow \varphi^{\mathrm{c0},n+1}, \mathbf{u}^{n+1}, p^{n+1}$ ▷ set previous time step to outdated t^{n+1}
  $\varphi^{\mathrm{c0},n+1,0} \leftarrow$ UpdateLevelSet($\mathbf{u}^n$, Δt) ▷ Algorithm 2
  $Agg(\mathfrak{K}_h^{\mathrm{X}}, A_\alpha^0) \leftarrow \mathfrak{K}_h^{\mathrm{X}}$ ▷ update agglomeration (Section 3.3.3)
  $\mathbf{M}^{n+1,0} \leftarrow$ ComputeMassMatrix() ▷ (Section 3.5.1)
  $\mathbf{u}^{n+1,0}, p^{n+1,0} \leftarrow$ Extrapolation($\mathbf{u}^n, p^n, \varphi^{\mathrm{c0},n+1}$)
    ▷ Initial solution on $\varphi^{\mathrm{c0},n+1}$ by Eq. (3.68)
  $\mathrm{Res}^0 \leftarrow$ computeResidual($\mathbf{u}^{n+1,0}, p^{n+1,0}$)
  while (!($\mathrm{Res}^i < \epsilon_{\mathrm{NSE}}$) && $i < i_{\max}$) **do** ▷ if implicit: !($\mathrm{Res}^i < \epsilon_{\mathrm{NSE}}$ && $\mathrm{Res}^i_{\mathrm{LS}} < \epsilon_{\mathrm{LS}}$)
    if (implicit interface evolution && $\mathrm{Res}^i < \epsilon_{\mathrm{NSE}}$) **then**
      $\varphi^{\mathrm{c0},n+1,i} \leftarrow$ UpdateLevelSet($\mathbf{u}^{n+1,i}$, Δt)
      $Agg(\mathfrak{K}_h^{\mathrm{X}}, A_\alpha^i) \leftarrow \mathfrak{K}_h^{\mathrm{X}}$
      $\mathbf{M}^{n+1,i} \leftarrow$ ComputeMassMatrix()
    end if
    $\mathbf{Op}^i, \mathrm{RHS}^i \leftarrow$ ComputeOperatorMatrixAndRHS($\mathbf{u}^{n+1,i}, p^{n+1,i}$) ▷ Eq. (3.38)
    $\mathbf{M}^{n+1,i,\alpha}, \mathbf{Op}^{i,\alpha}, \mathrm{RHS}^{i,\alpha} \leftarrow$ PerformAgglomeration($Agg(\mathfrak{K}_h^{\mathrm{X}}, A_\alpha^i)$)
    preconditioning ▷ (Section 3.9.1)
    BuildLinearSystem() ▷ according to time stepping scheme (Section 3.5.3)
    $\mathbf{u}^{n+1,i,\alpha}, p^{n+1,i,\alpha} \leftarrow$ SolveLinearSystem() ▷ sparse direct solver
    $\mathbf{u}^{n+1,i}, p^{n+1,i} \leftarrow$ RevertAgglomeration()
    $\mathbf{u}^* \leftarrow$ UpdateLinearization($\mathbf{u}^{n+1,i}$) ▷ Eq. (3.40)
    $\mathrm{Res}^i \leftarrow$ computeResidual($\mathbf{u}^{n+1,i}, p^{n+1,i}$)
    $i \leftarrow i + 1$
  end while
  PostProcessing(n,t^n,Δt)
end procedure

The algorithm for a single time step is described as pseudocode in Algorithm 5. First considering the solver run of a Stokes system, i.e. discarding the non-linear convective terms, the procedure starts by updating the level-set field φ^{c0}, resp. the interface position, by UpdateLevelSet (line 3). An initial solution on the new domain $Agg(\mathfrak{K}_h^{\mathrm{X}}, A_\alpha^0)$ is given by extrapolation of the old solution $\mathbf{u}^n$, p^n (line 6). Then, the linear system with corresponding mass and operator matrices and right-hand side (RHS) needs to be build (lines 14 to 17). The resulting system is preconditioned and solved by a linear solver, see Section 3.9.1. Note that the system is solved on the agglomerated polynomial space (line 15) and needs to be reverted after solving (line 19).

Considering the solution for the non-linear Navier-Stokes system an iterative process, e.g. Picard iteration, needs to be embedded until a prescribed convergence criterion ϵ_{NSE} is satisfied or a maximum number of iterations i_{max} is exceeded (line 8). In this case the convective terms (3.40) are linearized with the velocity of the previous iteration, i.e. $\mathbf{u}^* = \mathbf{u}^{n+1,i-1}$, and needs to be updated on every iteration (line 20).

Note that for steady-state calculations the solution is computed via an implicit Euler time stepping scheme, where the time step size is set to $\Delta t = 1 \cdot 10^{100}$ resulting in a vanishing contribution of the temporal term.

Coupling of the Flow Solver with the Interface Evolution There are two options for the coupling of the flow solver with the interface evolution: explicit and implicit. The procedure described above corresponds to an explicit coupling, i.e. the interface position is only updated once (line 3) before solving the Navier-Stokes equations. Thus, the new interface position is determined by the velocity field of the previous time step $\mathbf{u}^n$. This option is set as the default configuration. Computing the new interface position implicitly by the new velocity field $\mathbf{u}^{n+1}$, the level-set field φ^{c0} needs to be updated during the iteration process (line 9). Note that the update is not performed in every iteration, but only when the Navier-Stokes solution is converged. This way we guarantee that the interface evolution, resp. new position, does not disturb the convergence of the Navier-Stokes iterations by undesired errors of the velocity field due to high residuals. The coupled iteration process converges when both the Navier-stokes solution with ϵ_{NSE} and the level-set field with ϵ_{LS} converge.

3.9.1. Preconditioning and Solving the Saddle Point Problem

As described in Section 3.2.2 the discretized system can be written in matrix formulation, see Equation (3.17). In context of the (linearized) Navier-Stokes equations the corresponding operator matrix $\mathbf{Op}$ has the structure of a saddle point problem given as

$$\mathbf{Op} = \begin{bmatrix} \mathbf{Op}_{c,a} & \mathbf{Op}_b^{\mathrm{T}} \\ \mathbf{Op}_b & 0 \end{bmatrix}, \tag{3.132}$$

where $\mathbf{Op}_{c,a}$ denotes the convective trilinear form $c(-,-,-)$ (3.40) and viscous bilinear form $a(-,-)$ (3.42), and $\mathbf{Op}_b$ the bilinear forms for the pressure gradient and velocity divergence $b(-,-)$ (3.41). For small numbers of DOF such a system may be effectively solved using direct solver libraries like PARDISO (Schenk et al., 1999) or MUMPS (Amestoy et al., 2001; Amestoy

et al., 2006). Before actually solving the linearized system we perform the cell agglomeration (line 15) in order to reduce the condition number by removing small cut-cells. In context of the HMF quadrature rules, which may produce negative weights and thus negative norms in the mass matrix, this operation additionally reduces the number of such troubled cells. However, it is not guaranteed to remove all troubled cells. Therefore, a modified Cholesky decomposition is performed to remove all remaining cells with a negative norm. So far a positive definite mass matrix is guaranteed and the operator matrix is transformed to an orthonormal basis in the agglomerated XDG space. In a last stage the condition number is further reduced by block Jacobi preconditioning. This results in diagonal matrices containing only 0, -1 and +1 entries for the symmetric part of the block diagonal in the convection-diffusion block $\mathbf{Op}_{c,a}$. For more details on the respective matrix operations the reader is referred to Kummer (2016).

3.9.2. Adaptive Mesh Refinement

The XNSE-solver is extended to allow adaptive mesh refinement (AMR) during the simulation. In the context of DG methods this is implemented in a straightforward manner due to its locality property. The DG discretization allows the occurrence of hanging nodes between cells in the numerical mesh. Therefore, in order to locally refine the mesh one subdivides a cell into smaller cells. Within the BoSSS framework a cell is divided into four equal-sized sub-cells, which can be further subdivided. For higher refinement levels, we ensure that neighbouring cells also refine to always exhibit a 2:1 cell ratio on every edge. Thus, we counteract locking-effects on the refined cells by much larger cells.

The adaption of the new mesh is performed before RunSolverOneStep if predefined conditions for the refinement level on each cell are fulfilled. Note that each mesh adaption results in a soft-restart of the simulation run. For the XNSE-solver multiple AMR options are implemented: A constantInterface base refinement and additional refinements such as curvatureRefined, contactLineRefined and NavierSlipRefined. Examples of all options are displayed in Figure 3.11.

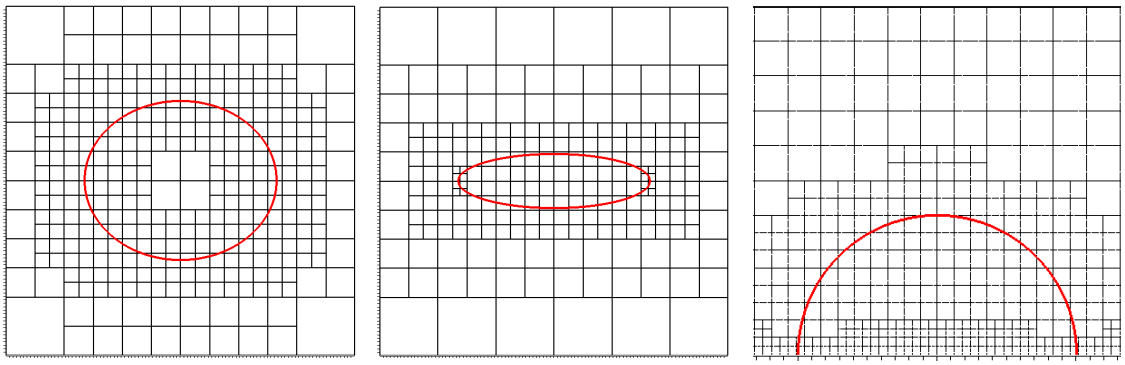


Figure 3.11.: Options for adaptive mesh refinement: (left) Constant base refinement with level 2 at the interface and neighbouring cells, (middle) additional refinement with level 3 in regions with high curvature, (right) additional refinement at the contact line and Navier-slip boundaries.

3.10. Conclusion

In this chapter main aspects of the developed multi-phase flow solver for transient sharp interface problems were presented. The spatial discretization is based on the extended Discontinuous Galerkin method proposed by Kummer (2016) for the steady Navier-Stokes equations. In this work the discretization was extended for the transient case. Therefore, the moving interface method developed by Kummer et al. (2018) was applied for the temporal discretization of the transient Navier-Stokes equations.
For the representation of the interface a high-order level-set method is used. In order to ensure continuity of the interface for the adaption of the discretization, a L^2-projection with continuity constraints was developed and implemented during this work. This method is applicable on meshes with hanging nodes and in three dimensions, and performs on multiple cores. Regarding the interface evolution, respectively the level-set advection, a fast-marching method processing on a narrow-band around the interface is used. For this algorithm the elliptic extension-velocity by Utz and Kummer (2018) and the elliptic reinitialization by Utz et al. (2017) were incorporated during this work. As an alternative method a Fourier-based explicit interface representation was developed and implemented for testing purposes in the early stages of development. However, this method is only applicable for wave-like and circular-shaped interfaces and thus not further discussed in the following Chapter 4.
Regarding the surface tension force different discretizations were implemented: the Laplace-Beltrami formulation, the semi-discrete formulation and a dissipative interface model, i.e. Boussinesq-Scriven model. Note that the Laplace-Beltrami formulation is considered as the standard option for the flow solver.

Throughout this chapter extensions for the numerical handling of contact line problems are presented. A main result of this work is the discretization of the generalized Navier-boundary condition for the XDG method. This involves various adaptions within the BoSSS-framework, such as the quadrature for the contact line in 3D. For both cases with and without contact line an energy estimate for the presented discretization is given.

The chapter closes with the overall structure of the developed XNSE-solver. In this work two strategies, i.e. explicit and implicit, for the coupling of the flow solver with the interface evolution were implemented. Furthermore, the solver was extended to allow various adaptive mesh refinement options. Again this involves the adaption of the quadrature in order to perform on cut-cells with hanging nodes.

4. Numerical Results of Transient Two-Phase Flows

In this chapter we present numerical results regarding the incompressible two-phase flow solver with a material interface and no contact lines at domain boundaries. The results demonstrate various features of the developed high-order sharp interface flow solver and serve as a thorough verification and validation basis. The presented test cases provide first a comparison to an analytical solution for capillary waves (Section 4.1). Then, numerical investigations are performed regarding the condition number, convergence rates and stability for a droplet assuming a fixed non-equilibrium state and equilibrium state in Section 4.2. The investigations considering the stability are extended for the dynamic case of an oscillating droplet, where additional energy issues are addressed (Section 4.3). The well-known rising bubble benchmark is considered as a last test case, where we compare our results to other two-phase flow solver (Section 4.4). In the end we conclude all results throughout this chapter in Section 4.5.

Error Quantification for Derived Scalar Measures Over Time In this chapter scalar measure properties, e.g. amplitude heights of waves, are often derived from the computed numerical DG-fields. In order to quantify the error against a suitable reference solution, the following error norms according to Hysing et al. (2009) are used

$$l_1 \text{ error} : \|e\|_1 = \frac{\sum_{n=1}^{\text{NTS}} \left|q_n - q_{n,\text{ref}}\right|}{\sum_{n=1}^{\text{NTS}} \left|q_{n,\text{ref}}\right|}, \tag{4.1a}$$

$$l_2 \text{ error} : \|e\|_2 = \left(\frac{\sum_{n=1}^{\text{NTS}} \left|q_n - q_{n,\text{ref}}\right|^2}{\sum_{n=1}^{\text{NTS}} \left|q_{n,\text{ref}}\right|^2} \right)^{\frac{1}{2}}, \tag{4.1b}$$

$$l_\infty \text{ error} : \|e\|_\infty = \frac{\max_n \left|q_n - q_{n,\text{ref}}\right|}{\max_n \left|q_{n,\text{ref}}\right|}, \tag{4.1c}$$

where q_n describes the corresponding numerical solution at time step t^n, i.e. $q_n = q(t^n)$ with $n = 1, ..., \text{NTS}$ and NTS denoting the total number of time steps. If no analytical reference solution is provided, the solution on the finest grid of a convergence study is used as reference. Linear interpolation is used, if there are less time steps than provided by the reference solution.

In a double-logarithmic representation the experimental order of convergence (EOC) is determined by the slope of a linear regression. The regression coefficients for a given set of logarithmic abscissas and values are estimated by an ordinary least-squares method. Further,

the rate of convergence (ROC) for a single refinement level l can be computed as (Hysing et al., 2009)

$$\text{ROC} = \frac{\log_{10}(\|e^{l-1}\| \,/\, \|e^{l}\|)}{\log_{10}(h^{l-1}/h^{l})}. \tag{4.2}$$

4.1. Capillary Wave

As a first test case we are considering the damped oscillations of a small-amplitude capillary wave, see Figure 4.1. The equilibrium state, i.e. a planar interface, is disturbed by a sinusoidal perturbation with an amplitude height of a_0 and wavelength of $\lambda = L$. In order to restore its equilibrium state the interface starts to oscillate due to the interfacial surface tension forces. This oscillatory behaviour is damped by viscous dissipation in the bulk phases. In the limit of small-amplitude initial perturbations, i.e. $a_0 \ll \lambda$, Prosperetti (1981) presents an exact analytical solution for the corresponding initial value problem. Such a problem allows to verify basic features of dynamic surface tension driven flows in context of viscous two-phase flows and is considered in many publications, e.g. Popinet and Zaleski (1999) and Popinet (2009).

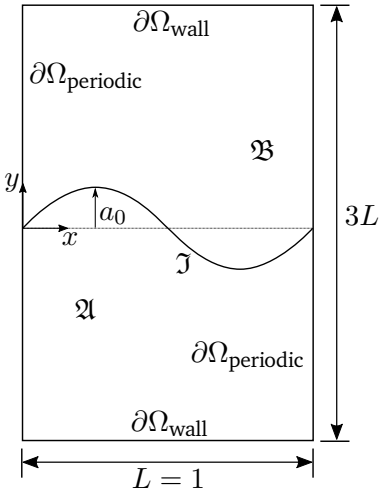


Figure 4.1.: Setup of the capillary wave test case with an initial sinusoidal perturbation with an amplitude height of a_0 and wavelength of $\lambda = L$.

Analytical Solution for the Initial Value Problem In the work of Prosperetti (1981) two viscous, incompressible fluids of infinite depth and lateral extent are superposed. In the following the subscript 'l' denotes the lower fluid and 'u' the upper one, respectively. Both fluids are governed by the linearized Navier-Stokes equations with the divergence free constraint of incompressibility, i.e. continuity equation. The interface position given as a standing wave in the (x, z)-plane reads

$$y = a(t)f(x, z), \tag{4.3}$$

where f fulfils the Helmholtz-equation $(\partial_x^2 + \partial_z^2 + k^2)f(x,z) = 0$ with $k = (k_x^2 + k_z^2)^{\frac{1}{2}}$. For a two-dimensional setting, as depicted above, the interface position reduces to

$$y = a(t)\sin(kx) \tag{4.4}$$

with $k = k_x = \frac{2\pi}{\lambda}$. An explicit formulation for the time-dependent amplitude height $a(t)$ can only be given analytically for the following two limiting cases: First, one fluid has a negligible dynamical effect, i.e. $\rho_\mathrm{l} = \mu_\mathrm{l} = 0$ or $\rho_\mathrm{u} = \mu_\mathrm{u} = 0$, and second, fluids have equal kinematic viscosities $\nu_\mathrm{l} = \nu_\mathrm{u}$. For both cases the amplitude height is given by

$$\begin{aligned} a(t) =& \frac{4(1-4\beta)\nu^2k^4}{8(1-4\beta)\nu^2k^4 + \omega_0^2} a_0 \mathrm{erfc}(\nu k^2 t)^{\frac{1}{2}} \\ &+ \sum_{i=1}^{4} \frac{z_i}{Z_i}\left(\frac{\omega_0^2 a_0}{z_i^2 - \nu k^2} - u_0\right) \mathrm{erfc}((z_i^2 - \nu k^2)t)\mathrm{erfc}(z_i t^{\frac{1}{2}}), \end{aligned} \tag{4.5}$$

where erfc() denotes the complementary error function, z_i with $i = \{1,2,3,4\}$ the four roots of the algebraic equation

$$\begin{aligned} &z^4 - 4\beta(k^2\nu)^{\frac{1}{2}}z^3 + 2(1-6\beta)k^2\nu z^2 \\ &\quad + 4(1-3\beta)(k^2\nu)^{\frac{3}{2}}z + (1-4\beta)\nu^2k^4 + \omega_0^2 = 0, \end{aligned} \tag{4.6}$$

and

$$\begin{aligned} Z_1 &= (z_2 - z_1)(z_3 - z_1)(z_4 - z_1), \\ Z_2 &= (z_1 - z_2)(z_3 - z_2)(z_4 - z_2), \\ Z_3 &= (z_1 - z_3)(z_2 - z_3)(z_4 - z_3), \\ Z_4 &= (z_1 - z_4)(z_2 - z_4)(z_3 - z_4). \end{aligned} \tag{4.7}$$

The dimensionless parameter β is zero for the one-fluid case and $\beta = \frac{\rho_\mathrm{l}\rho_\mathrm{u}}{(\rho_\mathrm{l}+\rho_\mathrm{u})^2}$ for the two-fluid case. The inviscid natural frequency ω_0 is given by

$$\omega_0 = \frac{\sigma k^3}{\rho_\mathrm{l} + \rho_\mathrm{u}}. \tag{4.8}$$

The values $a_0 = a(0)$ and $u_0 = \dot{a}(0)$ denote the initial perturbation height and velocity.

Comparison to the Analytical Solution We follow the approach of Popinet (2009) for comparing our numerical results with the analytical solution given above. The computational domain is set up as depicted in Figure 4.1 with $\Omega = [0, L] \times [-\frac{3L}{2}, \frac{3L}{2}]$. Thus, the lower and upper boundaries imposed by a wall boundary condition are far enough away from the interface. The wavelength of the sine wave is $\lambda = 1$ and the corresponding wave number is given by $k = 2\pi$. The initial perturbation height is set to $a_0 = \frac{\lambda}{100} = 0.01$ in a resting system, i.e. $u_0 = 0$.

In order to verify our numerical method for a representative range of physical regimes, a study of different Laplace numbers is investigated, i.e. $\mathrm{La} = \{3 \cdot 10^0, 1.2 \cdot 10^2, 3 \cdot 10^3, 3 \cdot 10^5\}$. The dimensionless Laplace number is defined as

$$\mathrm{La} = \frac{\sigma\rho L}{\mu^2} \tag{4.9}$$

and describes the ratio of surface tension and inertia forces to viscous dissipation in the bulk. The corresponding physical parameters are given in Table 4.1.

Table 4.1.: Physical parameters for the Laplace number study of capillary waves

La	ρ	μ	σ
3	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	$3 \cdot 10^{-3}$
$1.2 \cdot 10^{2}$	$1 \cdot 10^{-3}$	$5 \cdot 10^{-4}$	$3 \cdot 10^{-2}$
$3 \cdot 10^{3}$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-4}$	$3 \cdot 10^{-2}$
$3 \cdot 10^{5}$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-5}$	$3 \cdot 10^{-2}$

Note that we set the same density and dynamic viscosity values for both fluid phases, in order to comply with equal kinematic viscosities. All physical setups are computed on three different meshes with equidistant mesh sizes of $\frac{\lambda}{h} = \{8, 16, 32\}$. The time step sizes are set according to the capillary time step restriction (3.119) with a spatial discretization order of $k = 2$.

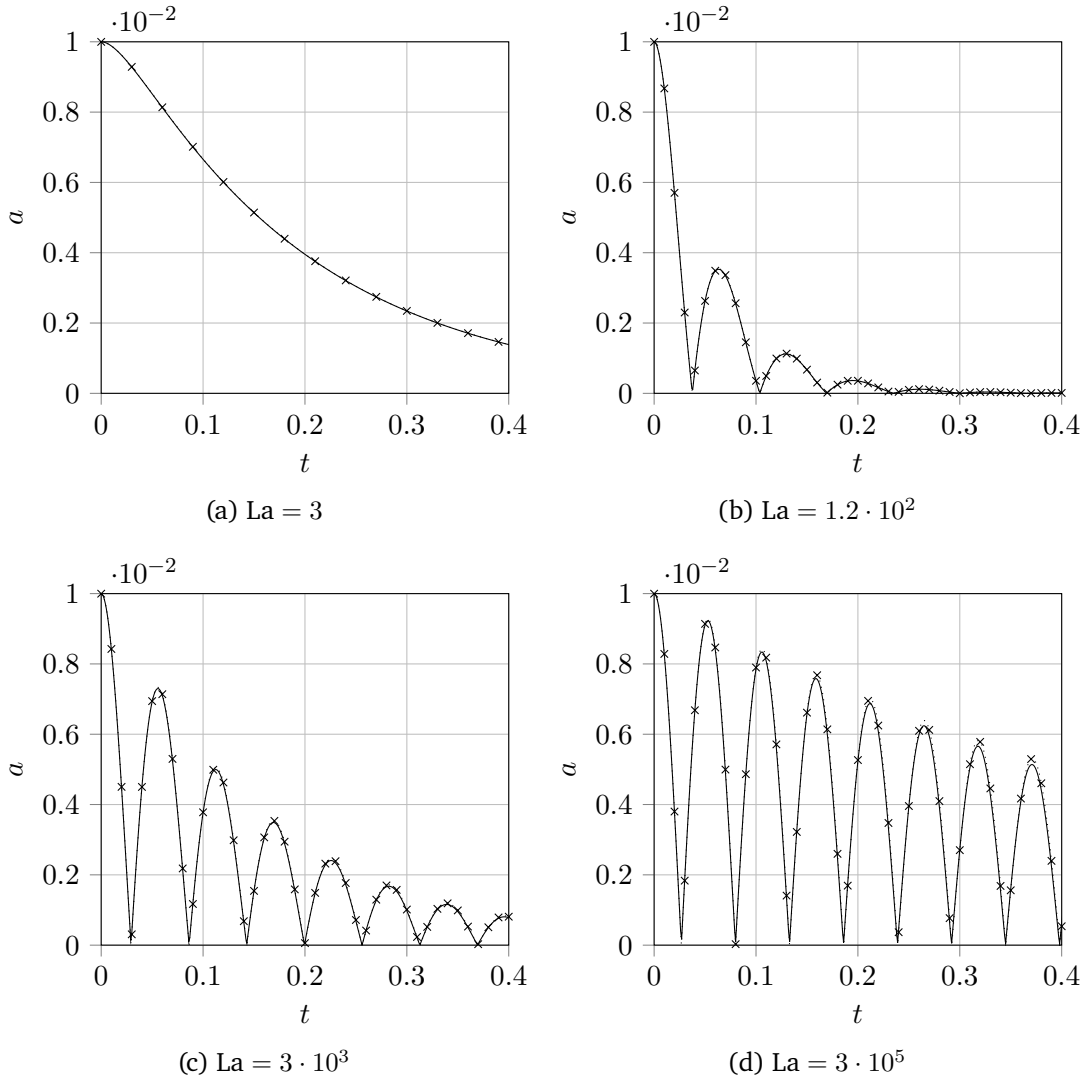
The numerical solutions for all Laplace numbers are given in Figure 4.2, where the solution on the finest grid is depicted. The analytical reference solutions (4.5) are computed using MATLAB, where the roots of equation (4.6) are numerically determined via `vpasolve()`.

One can see that the dynamical behaviour increases with increasing La. For low La the oscillations are damped due to the dominating viscous dissipation in the bulk phases. For higher La, i.e. higher surface tension forces, the damping rates decrease and the oscillation frequencies increase. All numerical results show very good agreement to the analytical solution (4.5), only for the highest La, i.e. most dynamic regime, it is visible that the numerical solution is underestimating the damping rate of the analytical solution. However, the oscillation frequency is still in very good agreement. The l_2-error norms for the whole mesh study are given in Table 4.2 with the corresponding EOC.

Table 4.2.: l_2-error norms for the capillary wave studies

La	$\frac{\lambda}{h} = 8$	$\frac{\lambda}{h} = 16$	$\frac{\lambda}{h} = 32$	EOC
3	$1.067 \cdot 10^{-}3$	$2.567 \cdot 10^{-}2$	$3.962 \cdot 10^{-}3$	2.268
$1.2 \cdot 10^{2}$	$2.954 \cdot 10^{-}1$	$1.051 \cdot 10^{-}2$	$3.487 \cdot 10^{-}3$	3.167
$3 \cdot 10^{3}$	$8.314 \cdot 10^{-}2$	$2.700 \cdot 10^{-}2$	$1.021 \cdot 10^{-}2$	1.495
$3 \cdot 10^{5}$	$1.713 \cdot 10^{-}1$	$7.827 \cdot 10^{-}2$	$2.084 \cdot 10^{-}2$	1.501
$3 \cdot 10^{5}$ ($\Delta t_{\text{fix}} = 4 \cdot 10^{-5}$)	$1.740 \cdot 10^{-}1$	$4.044 \cdot 10^{-}2$	$1.288 \cdot 10^{-}2$	1.856
$3 \cdot 10^{5}$ ($k = 3$, $\Delta t_{\text{fix}} = 2 \cdot 10^{-5}$)	$5.068 \cdot 10^{-}2$	$8.168 \cdot 10^{-}3$	$7.7893 \cdot 10^{-}3$	1.336

The observations in Figure 4.2 are emphasized by the error norms for the finest mesh with $\frac{\lambda}{h} = 32$, where the values increase with higher La. This is additionally attended by a decreasing EOC. Considering the most dynamic setup with La $= 3 \cdot 10^5$, a second mesh study is done with a fixed time step size Δt_{fix}, which is chosen to be smaller than the capillary time step for the finest mesh. The study is additionally performed for polynomial degree $k = 3$. The corresponding time step sizes and l_2-error norms are given in Table 4.2. For $k = 2$ one can see



(a) La $= 3$

(b) La $= 1.2 \cdot 10^2$

(c) La $= 3 \cdot 10^3$

(d) La $= 3 \cdot 10^5$

Figure 4.2.: Comparison of the amplitude a over time t between the numerical solution (dotted, marker set every 100th data point) and the analytical solution (4.5) (solid) for a capillary wave with La $= \{3, 1.2 \cdot 10^2, 3 \cdot 10^3, 3 \cdot 10^5\}$.

that the EOC is slightly improved and the error norms are reduced. This is in accordance with the results stated in Kummer et al. (2018), where quite small time steps are needed in order to obtain the desired convergence rates. This requires even smaller time steps for the higher polynomial degree $k = 3$ which shows a poorer EOC compared to $k = 2$. However, the error norms are smaller and one should note that the error norm of $k = 3$ on the coarsest mesh with $\frac{\lambda}{h} = 8$ is comparable to $\frac{\lambda}{h} = 16$ for $k = 2$.

In Figure 4.3 the numerical solutions on the finest mesh for the second study are displayed. Looking at the close-up of the last amplitude peak, it demonstrates the tendencies given in Table 4.2. Increasing the temporal resolution improves the agreement to the analytical solution in both frequency and damping rate. The agreement is superior with a higher polynomial approximation, which shows the ability of higher order methods. Note that even on the finest

mesh the oscillation of the interface is still confined to one cell. This demonstrates the sub-cell accuracy due to the high-order representation of the interface as a DG field.

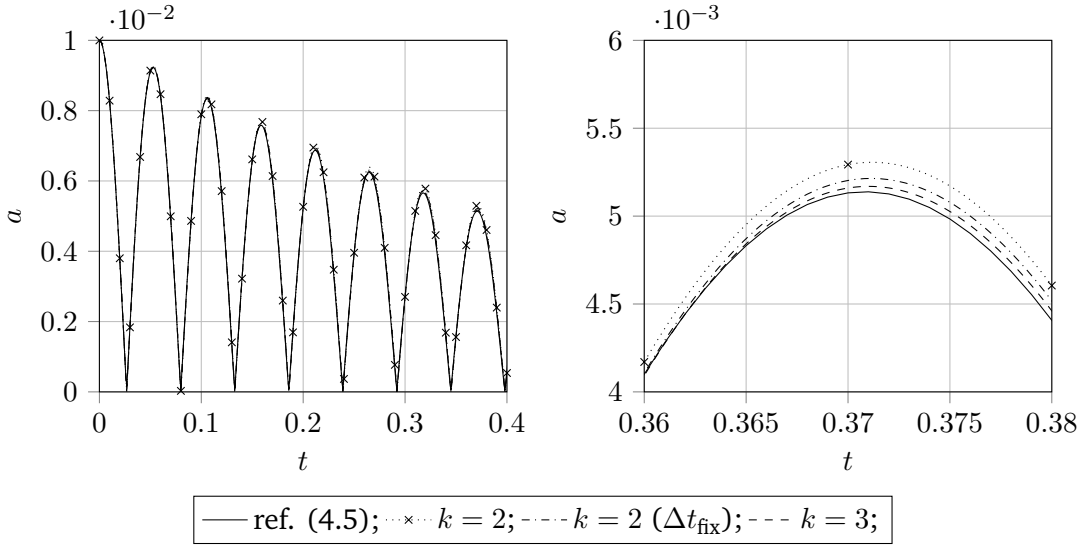


Figure 4.3.: Comparison of the numerical solutions for $k = 2$ ($\Delta t = 1 \cdot 10^{-4}$), $k = 2$ ($\Delta t_{\text{fix}} = 4 \cdot 10^{-5}$) and $k = 3$ ($\Delta t_{\text{fix}} = 2 \cdot 10^{-5}$) on the finest mesh with $\frac{\lambda}{h} = 32$ for $\text{La} = 3 \cdot 10^{-5}$.

4.2. Droplet in Steady-State

In this section two kinds of steady-state droplet simulations are investigated. The first one considers a droplet not in its circular equilibrium state. An artificial steady-state is achieved by switching off the level-set evolution and neglecting the temporal term, see Section 3.9. Thus, a spatial convergence study and condition number testing for a non-zero solution may be performed (Section 4.2.1). For the second case a circular droplet is presented, but for transient simulations. Thus, the accuracy and stability of the surface tension force computation is shown (Section 4.2.2).

4.2.1. Steady-State Simulations

For the steady-state simulations of a non-equilibrium droplet we follow the work of Kummer (2016) and extend some results and considerations. The droplet is fixed to an ellipsoidal shape given by

$$\varphi = \frac{x^2}{a^2} + \frac{y^2}{b^2} - 1, \tag{4.10}$$

where the semi-axes are set to $a = 0.816$ and $b = 0.784$. The droplet lies in the centre of the computational domain with $\Omega = [-1.5, 1.5] \times [-1.5, 1.5]$, where the boundary is described by the no-slip boundary condition. The physical parameters are: $\mu_{\mathfrak{A}} = 0.5$, $\mu_{\mathfrak{B}} = 0.05$, $\sigma = 0.1$

and both densities are set to unity, since at first only the Stokes-problem is considered. The solution for the described setting with a polynomial degree of $k = 4$ on a mesh with 144×144 cells is shown in Figure 4.4. The kink in the x-velocity field u is clearly visible on the left, and the sharp jump in the pressure field on the right due to the momentum jump condition (2.17).

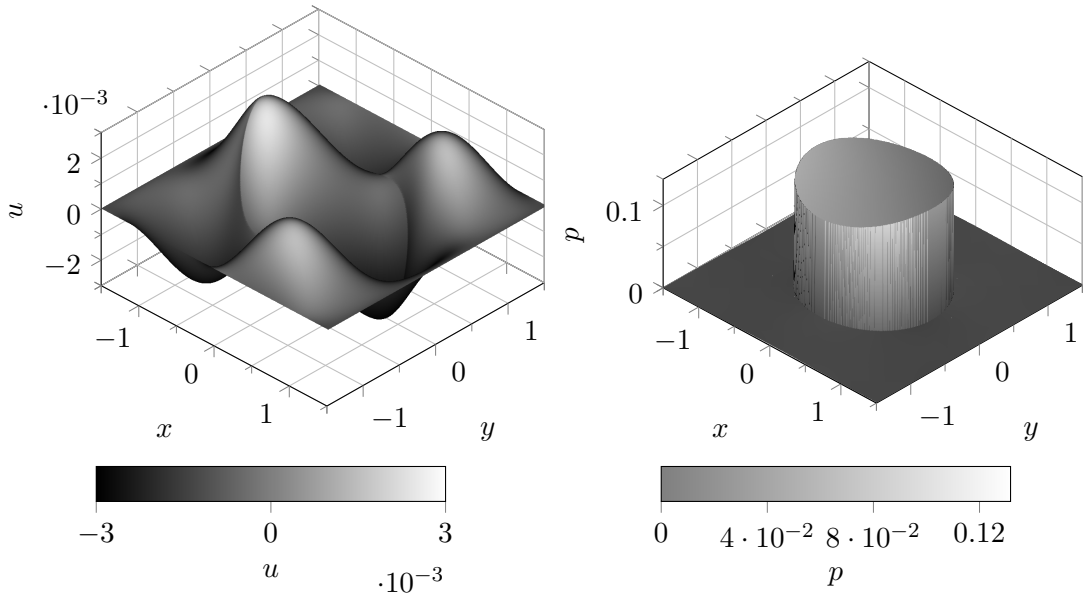


Figure 4.4.: x-velocity field u (left) and pressure field p (right) of a droplet not in equilibrium state for polynomial degree of $k = 4$ on a mesh with 144×144 cells.

Condition Number Test

The condition number of the operator matrix $\text{cond}(\mathbf{Op})$, i.e. in this case the Stokes-operator, is a measure for the numerical solvability of a linear system with $\mathbf{Op}(\tilde{\boldsymbol{\psi}}) = \mathbf{b}$. For small condition numbers relative small errors in the right hand side $\mathbf{b}$ only lead to relative small errors in the solution $\mathbf{x}$. This property is essential for the stability and accuracy of a numerical method. In context of iterative solver methods, a small condition number ensures that the reduction of the residual in each iteration also results in a convergent solution. For rigorous mathematical details the reader is referred to Meister (2015).

However, absolute condition number values are not sufficient to evaluate the solvability and stability of the discretized linear system. Thus, a mesh study is conducted for the steady-state droplet test case described above. The study with polynomial degree $k = 2$ is performed on meshes with $9 \times 9, 18 \times 18, 36 \times 36, 72 \times 72$ and 144×144 cells. The condition numbers are computed with MATLAB via `condest(A)` which computes a lower bound for the 1-norm condition number of a square matrix `A`. The results are shown in Figure 4.5 on the left.

Four sets of condition numbers are plotted. The values with the square marks denote the total condition number of the entire linear system. The corresponding line exhibits a slope of 1.85 w.r.t. to the 1-dimensional grid resolution. This means that the total condition number does

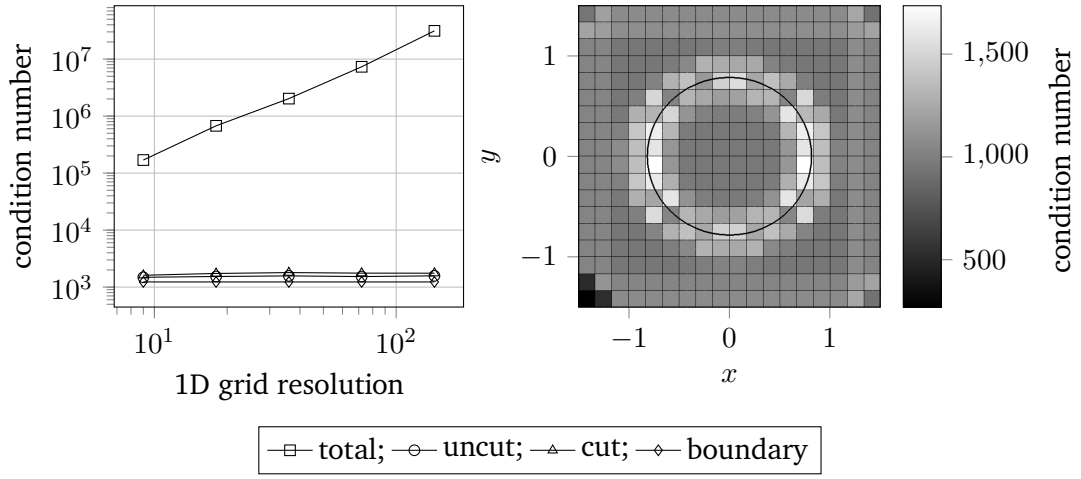


Figure 4.5.: Left: Condition number study for $k = 2$ of a steady-state droplet on the following meshes: $9\times9, 18\times18, 36\times36, 72\times72, 144\times144$. Considered are the total condition number of the entire system (square) and the maximum stencil condition number of inner uncut (circle) and cut cells (triangle) and boundary cells (diamond). Right: Stencil condition numbers on a 18×18 mesh.

not increase more than the size of the corresponding discretized operator matrix. For the 2-dimensional case we expect a maximum slope of 2, since the increase of the grid resolution in each dimension by a factor of 2 leads to a 4 times larger operator matrix. Exhibiting a slightly smaller value, the presented discretization does not introduce grid-dependent numerical artefacts and behaves as expected under grid refinement.
In contrast to this there are three sets of condition numbers, which are nearly constant for all grid resolutions. Those are local stencil condition numbers which denote the condition number for a cell in combination with its direct neighbours and considering all other cells as known. The maximum value form all cells is taken as the measure. For this study the set of all cells in the numerical mesh is divided into three subsets: boundary-cells (diamond marks), inner uncut-cells (circle marks) and inner cut-cells (triangle marks). All subsets show the expected behaviour of not varying for different grid sizes. Here, we expect that the discretization should be independent of the corresponding grid size and thus not influencing the local stencil condition number. In Figure 4.5 on the right the stencil condition numbers for all cells in a 18×18 mesh are shown. One should note that the cell local condition numbers for the cut-cells are roughly one order larger than both the inner uncut-cells and boundary-cells.
Additional condition number studies for polynomial degrees of $k = \{2, 3\}$ are found in Appendix A.1. Both studies show the expected behaviour as described above. But one should note that the absolute values increase for higher polynomial degrees.

Convergence Study

A spatial convergence study for the polynomial degrees $k = \{2, 3, 4\}$ is performed for the given base setting above on meshes with $9 \times 9, 18 \times 18, ..., 288 \times 288$ cells. The solution on the finest

mesh is used as reference solution for the error computation, i.e. solutions on the coarser meshes are injected onto the finest mesh in order to compute the L^2-error. The resulting error norms are presented in a double-logarithmic plot in Figure 4.6. The corresponding ROC values for each refinement level and the overall EOC value are given in Table 4.3. The EOC in all cases approximately matches the expected convergence behaviour of $k+1$, except for the study with $k=4$ ($k'=3$). In this case the EOC is still above the lower order studies, but not approaching the $k+1$ behaviour. However, regarding the corresponding values of the individual ROC, one should note that on finer meshes the values tend to the expected rates.

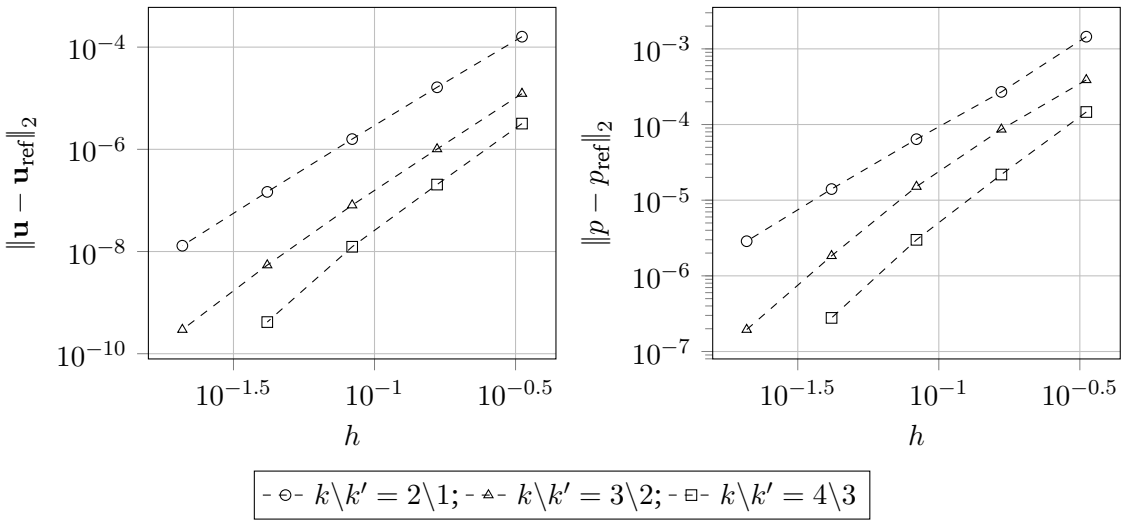


Figure 4.6.: Mesh convergence study of a steady-state droplet (non equilibrium state) for $k=\{2,3,4\}$ on meshes with $9\times 9, 18\times 18, ..., 288\times 288$ cells.

Table 4.3.: Convergence rates (ROC and EOC) for the mesh convergence study of a steady-state droplet for $k=\{2,3,4\}$.

$\frac{1}{h}$	18	36	72	144	EOC
ROC for $\|\mathbf{u}-\mathbf{u}_{\text{ref}}\|_2$					
$k=2$	3.29	3.37	3.43	3.5	3.40
$k=3$	3.60	3.65	3.90	4.19	3.82
$k=4$	3.97	4.04	4.90	-	4.28
ROC for $\|p-p_{\text{ref}}\|_2$					
$k'=1$	2.42	2.08	2.19	2.29	2.22
$k'=2$	2.17	2.51	3.04	3.24	2.75
$k'=3$	2.74	2.87	3.43	-	3.00

4.2.2. Transient Simulations

In this section the transient simulation of a droplet initialized by its equilibrium state, i.e. circular with a zero velocity field, is investigated. According to the momentum balance

equation (2.17) the corresponding pressure inside a droplet with radius r is given by

$$p_{\mathfrak{A}} = p_{\mathfrak{B}} + \frac{\sigma}{r}, \tag{4.11}$$

where $\frac{1}{r}$ describes the curvature in $\Omega \in \mathbb{R}^2$. Equation (4.11) is also referred to as the Young-Laplace law. Due to certain numerical inaccuracies the numerical solution does not provide a zero velocity field up to round-off errors, see Section 3.7. To quantify the discretization error Smolianski (2001) proposed the following test setup. Considered is a circular droplet with radius $r = 0.25$ in the centre of a unit square which we set to $\Omega = [-0.5, 0.5] \times [-0.5, 0.5]$. The boundary imposes a no-slip boundary condition. Both phases share the same properties with: $\rho = 10^4$, $\mu = 1$ and $\sigma = 1$. The corresponding Laplace number (Equation (4.9)) with $L = 2r$ is given as $\mathrm{La} = 5 \cdot 10^3$. The simulation is run until $t = 125$ with a fixed time step of $\Delta t = 0.01$. We investigate a mesh study with the following mesh sizes $\frac{1}{h} = \{20, 40, 60, 80\}$. In Table 4.4 the L^2-error norms for the spurious velocities and against the exact Young-Laplace solution are given. The computation is performed on the terminal time step at $t = 125$.

Table 4.4.: L^2-error norms for spurious velocities and Laplace-Young equation.

$\frac{1}{h}$	20	40	60	80
$\|\mathbf{u}\|_{L^2}$	$1.68 \cdot 10^{-5}$	$2.95 \cdot 10^{-7}$	$2.60 \cdot 10^{-7}$	$1.34 \cdot 10^{-7}$
$\|p - p_{\text{exact}}\|_{L^2}$	$1.03 \cdot 10^{-2}$	$5.20 \cdot 10^{-4}$	$5.68 \cdot 10^{-4}$	$4.15 \cdot 10^{-4}$

Comparing the numerical solution to the exact solution, it is remarkable that the error norms seem to converge already on the second mesh. This may results from the fixed time step size and long simulation time. In Figure 4.7 we take a closer look on the finest solutions by evaluating the temporal evolution of the kinetic energy norm $\left\| \frac{\rho}{2} \mathbf{u} \cdot \mathbf{u} \right\|_{L^2}$ in the bulk.

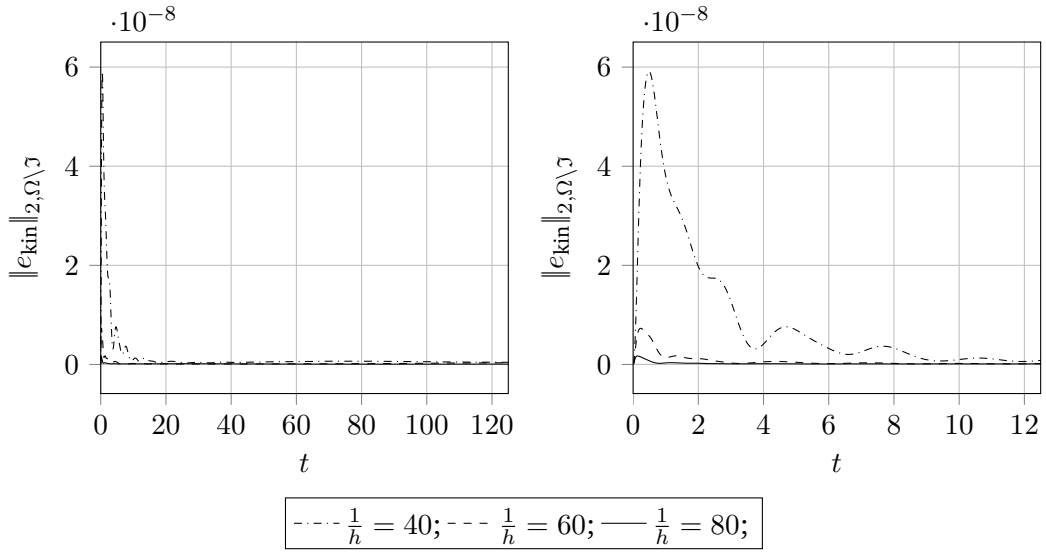


Figure 4.7.: Temporal evolution of the kinetic energy in both bulk phases for the presented mesh study with $\frac{1}{h} = \{40, 60, 80\}$. On the right hand side the range is restricted to $t = [0, 12.5]$.

One sees that for the coarser meshes the initial deviation from a zero-velocity field is much larger compared to the finest solution. Further, the decay of the spurious velocities takes a longer period of time. Additionally, another observation is that the temporal evolution resembles a damped oscillation. The damping is due to the viscous dissipation in the surrounding bulk phases, where the corresponding evolution is shown in Figure 4.8 on the left. Note that the viscous dissipation is always negative, which demonstrates the energetic stability of the presented discretization in the bulk phases, see Section 3.8.

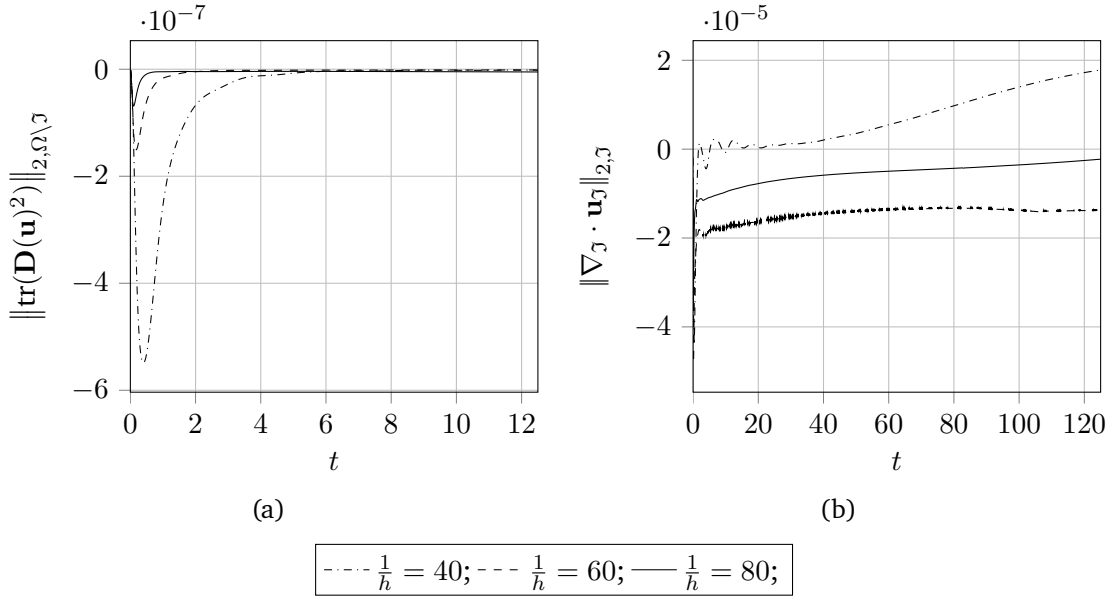


Figure 4.8.: Temporal evolution of the viscous dissipation in both bulk phases (a) and surface divergence (b) for the presented mesh study with $\frac{1}{h} = \{40, 60, 80\}$.

Besides the viscous dissipation the change rate of the interface area described as the surface divergence $\nabla_{\mathfrak{I}} \cdot \mathbf{u}_{\mathfrak{I}}$ is given in Figure 4.8. Here, one should note that all simulations still exhibit an enlarging or diminishing interface area until the end. For the coarsest mesh the change rate is even increasing.

4.3. Oscillating Droplet

In this section a basic dynamic test case for two-phase flow solver is presented. Here, we consider an initially perturbed droplet that starts to oscillate in order to restore its circular equilibrium state due to surface tension forces. As initial droplet shape an ellipsoidal form is taken as given by Equation (4.10).

As a first setup, we follow the configurations in Hysing (2006). The computational domain is set to $\Omega = [-0.5, 0.5] \times [-0.5, 0.5]$ with no-slip boundary condition imposed on all boundaries. The elliptic shape (initial radius $r = 0.25$) is given with $a = 1.25$ and $b = 0.8$. As for the capillary wave test case, both phases inhibit the same physical properties: density $\rho = 10^4$, dynamic viscosity $\mu = 1$, and a surface tension coefficient with $\sigma = 0.1$. This results in a

Laplace number of La ≈ 500. The simulations run until $t = 1000$ and are performed on successive finer meshes with $\frac{1}{h} = \{10, 20, 40, 60, 80\}$ and a fixed time step size of 0.5. Using a polynomial degree of $k = 2$, the time step restriction (3.119) is already exceeded on the second mesh, see Table 4.5.

Table 4.5.: Capillary time step restrictions for the mesh study of an oscillating droplet.

$1/h$	10	20	40	60	80
Δt_σ	1.086	0.384	0.136	0.074	0.048
$\Delta t/\Delta t_\sigma$	0.5	1.3	3.7	6.8	10.4

Note that in comparison to Hysing (2006), we use a smaller time step size due to the adapted grid size in (3.119) and a reduced range of mesh refinement level due to the CFL-condition (Lewy et al., 1928) for the explicit level-set advection.

In Figure 4.9 the evolution of the semi-axis in x-direction is plotted over the simulation time. It is remarkable that all simulations of the study show a stable oscillating behaviour until the end, see Figure 4.9 on the right. Note that on the finest mesh the capillary time step restriction is exceeded by a factor of 10.

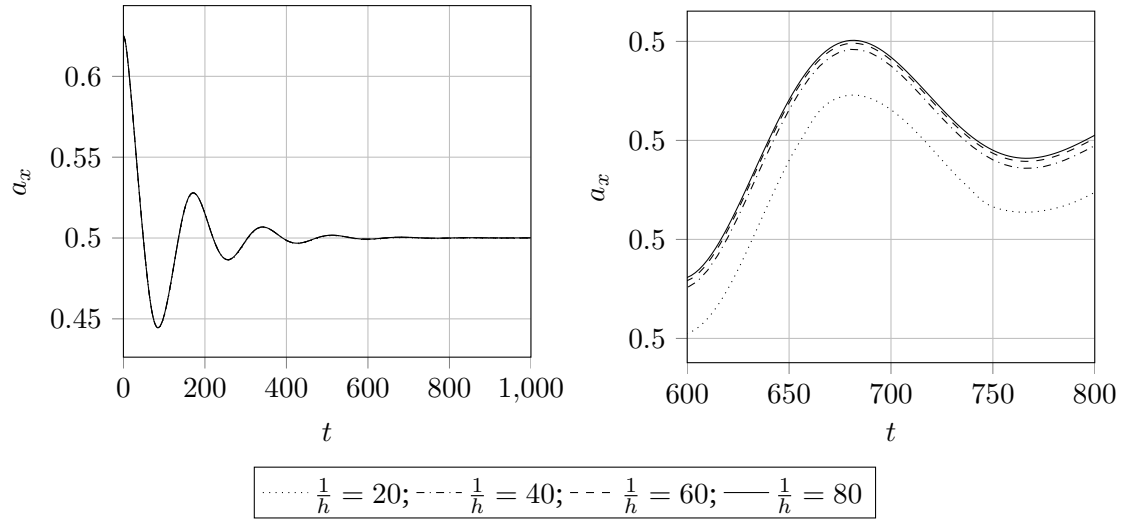
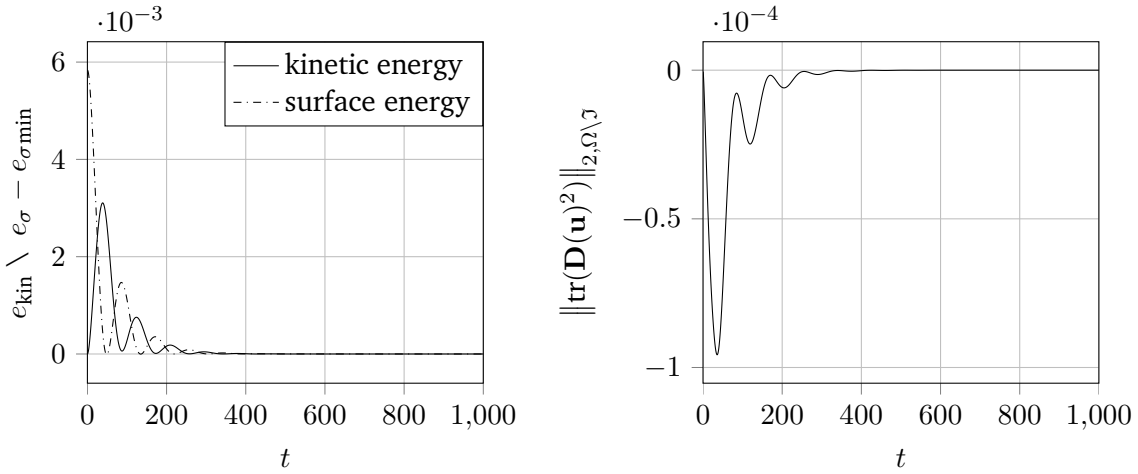


Figure 4.9.: Temporal evolution of the semi-axis a_x in x-direction for the presented mesh study of an oscillating droplet.

Taking a closer look at the finest solution, the temporal evolution of the kinetic energy in the bulk, i.e. $e_{\text{kin}} = \int_{\Omega\setminus\mathfrak{I}} \frac{\rho}{2}(\mathbf{u}\cdot\mathbf{u})\,\mathrm{d}V$, and of the surface energy, i.e $e_\sigma = \sigma \oint_\mathfrak{I} 1\,\mathrm{d}S$, is displayed in Figure 4.10. Note that the minimal surface energy, i.e. $e_{\sigma\min} = e_\sigma|_{t=1000}$, is subtracted from the plotted surface energy values. One can observe the transfer between the initially disturbed surface energy and the kinetic energy in the surrounding bulk phases. The maximum values of the surface energy corresponds directly to the minimum value of the kinetic energy. Here, the droplet takes the largest extend along one semi-axis, before it tends to restore its circular equilibrium state. Between these states the kinetic energy reaches its maximum always slightly before the minimum value of the surface energy. Due to the high Laplace number the

oscillations are strongly damped by viscous dissipation in the bulk. The temporal evolution of this quantity is shown in Figure 4.10 on the right. The oscillatory behaviour corresponds to that of the kinetic energy. Note that the viscous dissipation term again is always negative as discussed in Section 4.2.1. In this case the energetic stability is also shown for transient simulations including deformations of the interface.



(a) (b)

Figure 4.10.: (a) Temporal evolution of the kinetic energy in the bulk and surface energy for the finest mesh solution, $\frac{1}{h} = 80$. (b) Viscous dissipation in the bulk over the simulation time for the finest solution.

4.4. Rising Bubble Benchmark Test Case

In this section we present numerical results for the well known rising bubble benchmark for two-dimensional incompressible interfacial flow solver established by Hysing et al. (2009). In this setup the rise of a bubble in a liquid column is considered. The initial setting is depicted in Figure 4.11. A circular bubble with radius $r_0 = 0.25$ is positioned at $(0.5, 0.5)$ in the computational domain of $\Omega = [0, 1] \times [0, 2]$. The lower and upper boundaries are imposed with a no-slip boundary condition, i.e. $\mathbf{u} = 0$, and the left and right boundaries with a free-slip boundary condition given by

$$\mathbf{u} \cdot \mathbf{n}_{\partial\Omega} = 0, \quad \boldsymbol{\tau} \cdot \left(\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathrm{T}}\right) \mathbf{n}_{\partial\Omega} = 0, \tag{4.12}$$

where $\boldsymbol{\tau}$ denotes the tangent vector on the boundary $\partial\Omega$. The driving force of this benchmark is the gravity g that is directed in negative y-direction. Due to the difference in density, i.e. $\rho_{\mathfrak{A}} < \rho_{\mathfrak{B}}$, the bubble moves toward the upper boundary. Depending on the physical parameter setting the bubble deforms more or less on its way upward. For this setting two different test cases are defined. The corresponding physical parameters are given in Table 4.6.

Table 4.6.: Physical parameters for both test cases of the rising bubble benchmark.

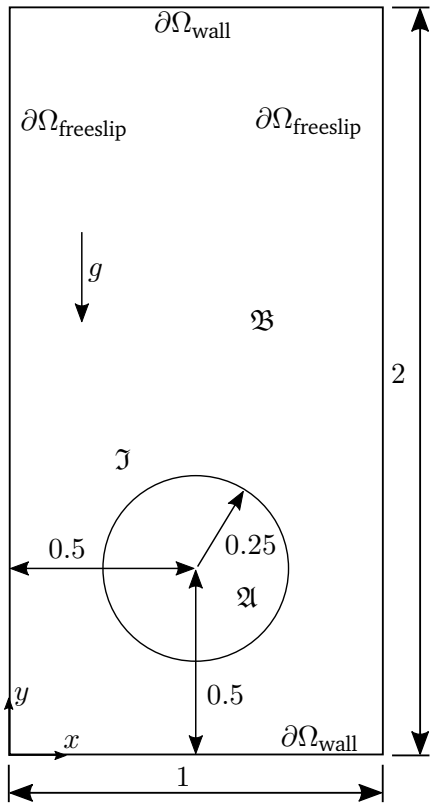


Figure 4.11.: Setup of the rising bubble benchmark.

Test case	$\rho_{\mathfrak{A}}$	$\rho_{\mathfrak{B}}$	$\mu_{\mathfrak{A}}$	$\mu_{\mathfrak{B}}$	σ	g	Eo
1	100	1000	1	10	24.5	0.98	10
2	1	1000	0.1	10	1.96	0.98	125

The first setup with density and viscosity ratios of 10 yields an Eötvös number of Eo $= 10$. The Eötvös number describes the ratio of gravitational forces to surface tension effects and is defined as follows

$$\text{Eo} = \frac{\rho_{\mathfrak{B}} U_g^2 2r_0}{\sigma} \tag{4.13}$$

with U_g denoting the gravitational velocity with $U_g = \sqrt{g2r_0}$. For the first test case the bubble shape ends in a stable ellipsoidal shape. The simulation is run for 3 time units. For the second test case with a decreased surface tension coefficient and higher Eo the bubble exhibits a more non-convex shape with filaments, which eventually break off. The main bubble assumes the terminal shape of a dimpled cap. Again the simulation is run for 3 time units.

Benchmark Quantities In order to rigorously compare the different numerical methods Hysing et al. (2009) introduces three scalar measures for the temporal evolution of the rising bubble. First, the centre of mass for tracking the translation of the bubble is considered, which is given as

$$\mathbf{x}_c = \frac{\int_{\mathfrak{A}} \mathbf{x} \, \mathrm{d}V}{\int_{\mathfrak{A}} 1 \, \mathrm{d}V}. \tag{4.14}$$

The second measure is denoted as circularity and defined by

$$\not\epsilon = \frac{\text{perimeter of area-equivalent circle}}{\text{perimeter of bubble}}. \tag{4.15}$$

The maximum value of the circularity is given for a perfectly circular bubble, i.e. $\not\epsilon = 1$, and is smaller for deformed bubble shapes. The third measure is the mean bubble velocity with

$$\mathbf{u}_\mathrm{c} = \frac{\int_{\mathfrak{A}} \mathbf{u} \, \mathrm{d}V}{\int_{\mathfrak{A}} 1 \, \mathrm{d}V}, \tag{4.16}$$

where the component in y-direction, i.e. in opposite direction to the gravitational vector, is denoted as the rise velocity V_c. The error quantification for these quantities is done via the error norms (4.1a) to (4.1c) defined in the beginning of this chapter. Besides these three quantities the terminal shape of the bubble at $t = 3$ is compared.

Benchmark Groups for Comparison In Hysing et al. (2009) three research groups participate to provide an extensive data-set to compare with. We give a brief overview on the groups methodology. The first group's code (TP2D, Transport Phenomena in 2D) is based on a finite-element discretization using the level-set method for the interface treatment. The flow variables velocity and pressure are solved decoupled with a discrete projection method. The interface contour is approximated by straight lines and the direct integration over these incorporates the surface tension effects. The second group (FreeLIFE, Free-Surface Library of Finite Element) also adopt the level-set approach. The spatial discretization is based on a piecewise linear finite-element approach. Like the first group, an additional global mass correction step is applied, see Equation (3.94). The last group (MooNMD, Mathematics and object-oriented Numerics in MagDeburg) uses inf-sup isoparametric finite elements for the solution of the incompressible Navier-Stokes equations. The arbitrary Lagrangian-Eulerian approach is used treating two-phase flows and no mass correction step is applied. The surface tension forces are computed via the Laplace-Beltrami operator, see Section 3.7. Further details are given in the benchmark paper Hysing et al. (2009).

4.4.1. Test Case 1

For the first test case a mesh study with $\frac{1}{h} = \{10, 20, 40, 60, 80\}$ is presented. The study is performed with polynomial degrees of $k = \{2, 3\}$ for the spatial XDG discretization and for both the BDF3-scheme is chosen in time. The time step sizes are set such that the corresponding capillary time step restriction (3.119) is satisfied. The total number of DOF for each performed setting are given in Table 4.7

Table 4.7.: Total Number of DOF for the rising bubble benchmark test case 1.

$\frac{1}{h}$	10	20	40	60	80
$k = 2$	-	12000	48000	108000	192000
$k = 3$	5200	20800	83200	187200	-

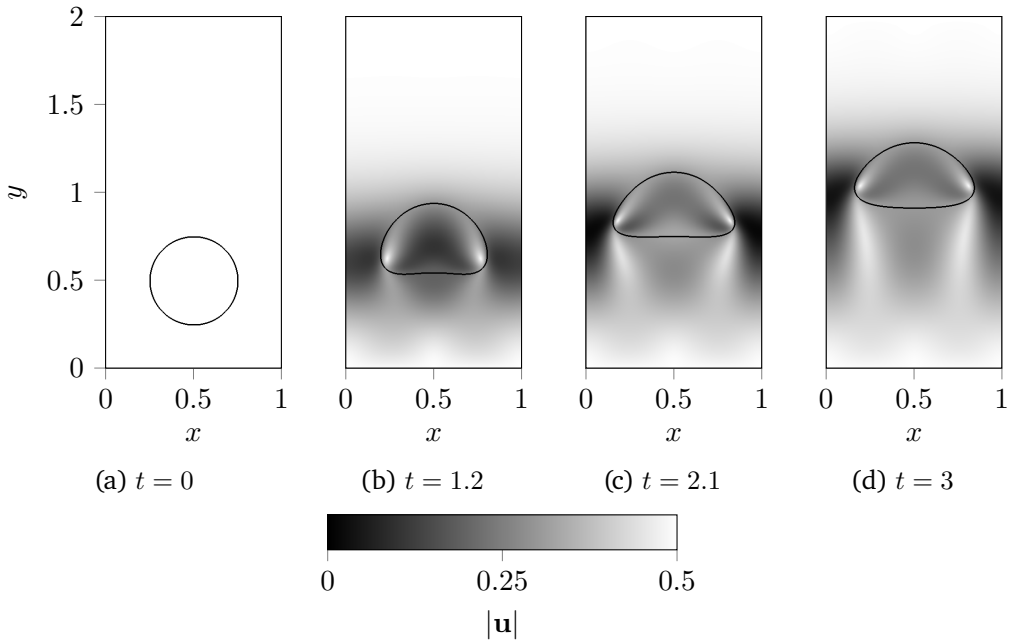


Figure 4.12.: Temporal evolution of the rising bubble benchmark test case 1 for times $t = \{0, 1.2, 2.1, 3\}$. The shown results are computed with a polynomial degree of $k = 3$ on a mesh with $\frac{1}{h} = 60$. The plotted field describes the magnitude of the velocity vector field $|\mathbf{u}|$.

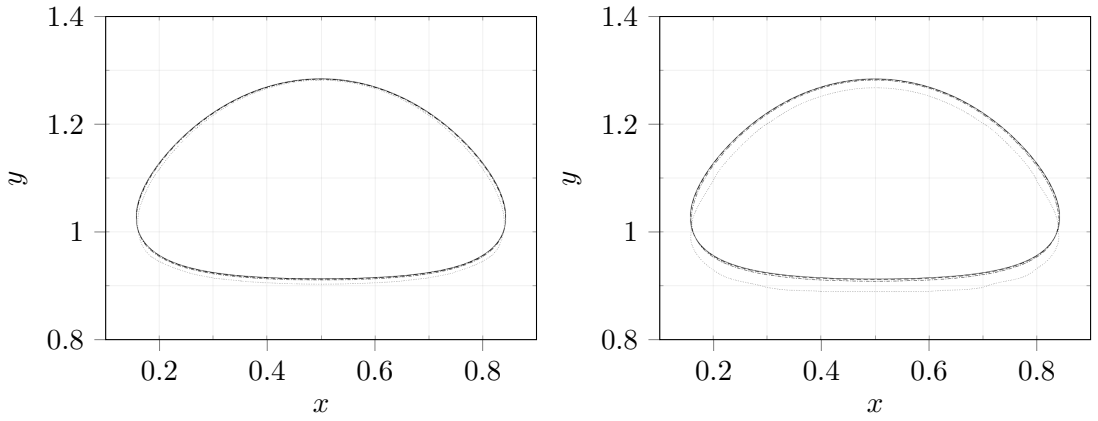
In Figure 4.12 the temporal evolution of the bubble for the simulation with $k = 3$ on the finest mesh $\frac{1}{h} = 60$ is displayed for the times $t = \{0, 1.2, 2.1, 3\}$. The plotted field is the magnitude of the corresponding velocity vector field.

The bubble forms a dimple on the lower side in the beginning, see $t = 1.2$, which recedes during its rise upward. In the end the bubble forms a stable ellipsoidal shape as described above. The convergence of the terminal shape at $t = 3$ is shown in Figure 4.13 for both polynomial degrees $k = \{2, 3\}$.

In both cases the convergence of the second mesh refinement is already very good. The agreement between the solutions on the upper side is even better throughout the mesh study. Thus, the deviation on the lower side clearly indicates an increase of the initial bubble area for lower resolutions.

In Figure 4.14 the temporal evolution of the benchmark quantities and the bubble area are plotted for the mesh study of $k = 2$. The results for $k = 3$ are found in Appendix A.2. As indicated in Figure 4.13 the mass production reduces with finer resolutions, where for the finest mesh the terminal relative mass production is $1.356 \cdot 10^{-3}$ for $k = 2$ and $1.24 \cdot 10^{-3}$ for $k = 3$.

Looking at the other benchmark quantities the convergence after the first refinement is very good showing no distinguishable deviations, except for the circularity after its minimum at around $t = 1.9$.



(a) $k = 2$ for $\frac{1}{h} = \{20, 40, 60, 80\}$ (b) $k = 3$ for $\frac{1}{h} = \{10, 20, 40, 60\}$

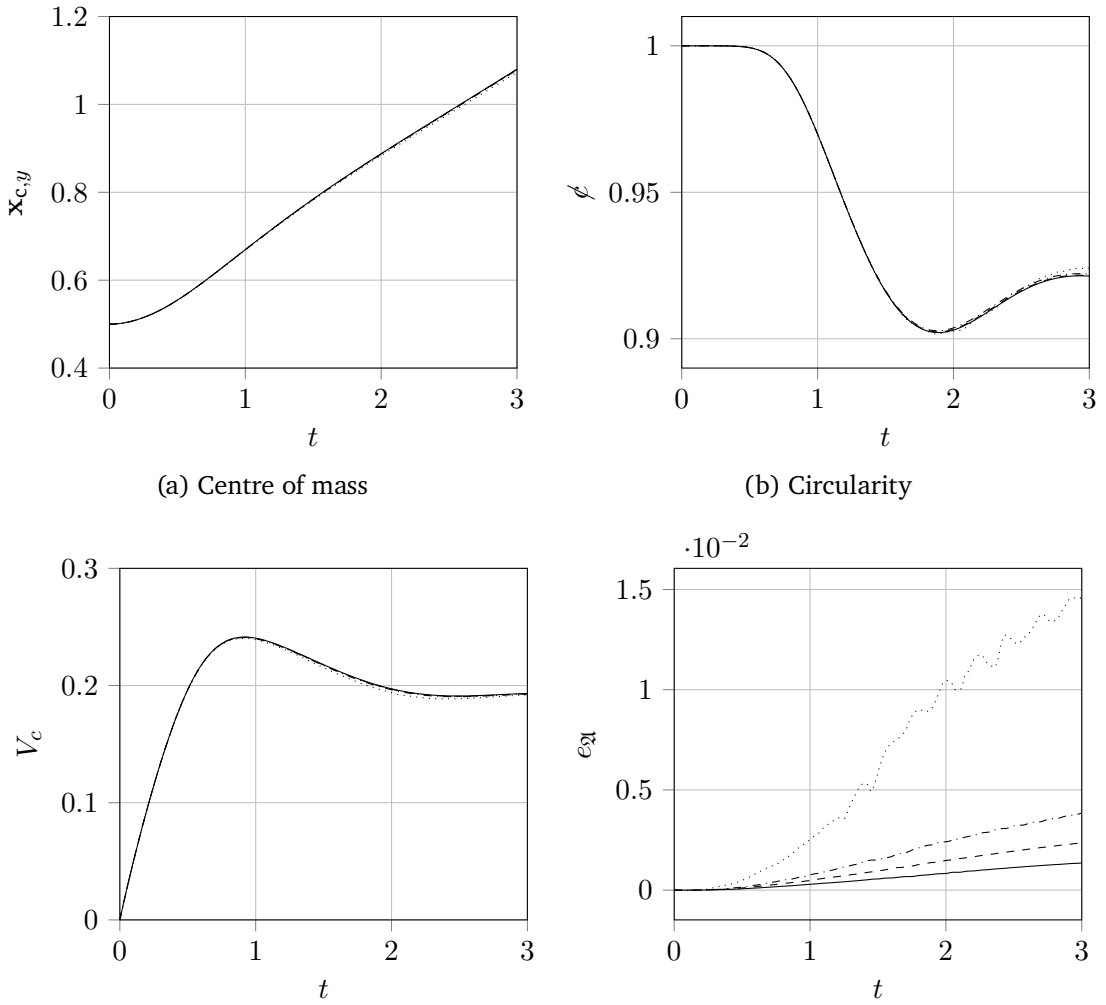
Figure 4.13.: Terminal bubble shapes for test case 1 of the mesh study. For both plots the following legend applies: 1. mesh (dotted), 2. mesh (dashdotted), 3. mesh (dashed), 4. mesh (solid).

The l_∞-error norms for the benchmark quantities against the finest resolution are given in Table 4.8 for $k = \{2, 3\}$. Additional, the corresponding ROC for each refinement are presented. The l_1-error and l_2-error norm are found in Appendix A.1 and A.2.

Table 4.8.: l_∞-error norms and ROC_∞ for the mesh study of the rising bubble benchmark.

	Center of mass		Rise velocity		Circularity	
$\frac{1}{h}$	$\|e\|_\infty$	ROC_∞	$\|e\|_\infty$	ROC_∞	$\|e\|_\infty$	ROC_∞
$k = 2$						
20	$6.67 \cdot 10^{-3}$	-	$1.43 \cdot 10^{-2}$	-	$2.61 \cdot 10^{-3}$	-
40	$1.74 \cdot 10^{-3}$	1.94	$3.50 \cdot 10^{-3}$	2.03	$8.57 \cdot 10^{-4}$	1.61
60	$6.27 \cdot 10^{-4}$	2.52	$1.20 \cdot 10^{-3}$	2.64	$3.12 \cdot 10^{-4}$	2.49
$k = 3$						
10	$2.05 \cdot 10^{-2}$	-	$2.06 \cdot 10^{-2}$	-	$2.04 \cdot 10^{-3}$	-
20	$2.95 \cdot 10^{-3}$	2.80	$7.38 \cdot 10^{-3}$	1.48	$2.59 \cdot 10^{-3}$	-0.34
40	$9.17 \cdot 10^{-4}$	1.69	$2.07 \cdot 10^{-3}$	1.83	$7.71 \cdot 10^{-4}$	1.40

The ROC of all benchmark quantities for $k = 2$ provide a high-order convergence up to 2.5. Compared to that the values for $k = 3$ shows poorer results with ROC below 2. However, this result agrees very well with the results of Kummer et al. (2018) regarding the moving interface approach (Section 3.5). Although computed with a smaller time step size, the temporal discretization with BDF3 does not match the theoretical requirement for a spatial discretization with $k = 3$, i.e. a time integration scheme of at least $2k$. Thus for this setting, we cannot expect the higher-order convergence rates. However, an increase of the spatial discretization order still produces smaller relative norms on the same mesh sizes. Note that a reinitialization is performed every 50th time step to stabilize the interface on the coarsest mesh for $k = 3$. This additional local operation leads to the seemingly better error norm for the circularity (4.15). The circularity interpreted as an indicator for the shape of the bubble strongly depends on the spatial resolution and local inaccuracies. Both other properties, centre



(a) Centre of mass

(b) Circularity

(c) Rise velocity

(d) relative error mass

Figure 4.14.: Convergence study for $k = 2$ of the temporal evolution of the benchmark quantities: (a) y-component of the centre of mass $\mathbf{x}_c$ (4.14), (b) circularity $\not\epsilon$ (4.15), (c) rise velocity V_c and (d) relative error of the bubble area $e_{\mathfrak{A}} = \left| \frac{\int_{\mathfrak{A}} 1 \, \mathrm{d}V - 2\pi r_0^2}{2\pi r_0^2} \right|$.

of mass and the rise velocity, are integral measures that do not strongly account for point-wise errors.

Comparison to benchmark groups

The results for the finest mesh of the presented mesh study with $k = \{2, 3\}$ in BoSSS are compared against the solutions for the finest mesh of the benchmark groups in Hysing et al. (2009). The corresponding mesh sizes h and total number of DOF are given in Table 4.9.

Table 4.9.: Spatial resolution and total number of DOF of the finest solutions for the rising bubble benchmark (test case 1).

	BoSSS		TP2D	FreeLIFE	MooNMD
	$k=2$	$k=3$			
$\frac{1}{h}$	80	60	320	160	900 (NDOF_{int})
NDOF	192000	187200	1231681	219765	72836

Beginning with the terminal shapes in Figure 4.15 no significant differences are visible throughout all methods. Even zooming into the leading and trailing edge at the bubble centre line shows very good agreement between all solutions.

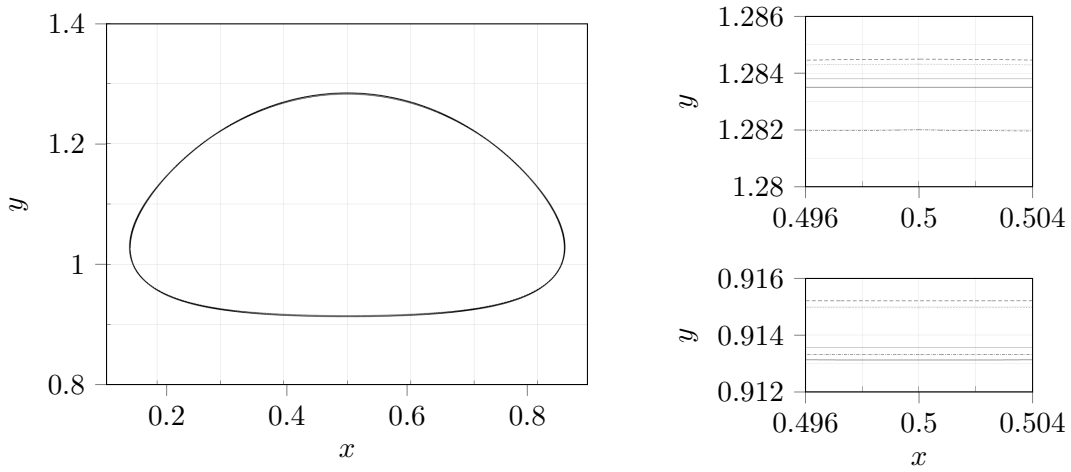


Figure 4.15.: Comparison of the terminal bubble shapes for test case 1: $k=2$ on mesh $\frac{1}{h}=80$ (solid gray), $k=3$ on mesh $\frac{1}{h}=60$ (solid black), TP2D (dotted), FreeLIFE (dashdotted) and MooNMD (dashed). On the right: Zoom at the leading (upper) and trailing edge (lower).

The comparison for the temporal evolution of the benchmark quantities are presented in Figure 4.16 to Figure 4.18.

Starting with the centre of mass (Figure 4.16), the overall agreement shows superior agreement, only a zoom to the end $t=[2.75,3]$ shows two distinct trends. Our numerical solutions for both polynomial degrees coincide very well with the second group (FreeLIFE). Likewise, the other two benchmark groups coincide, but are constantly ahead of our results. The terminal rise height at $t=3$ for both mesh studies are found in Appendix A.3 and A.4. Additional, the range of values provided by the finest solution of the benchmark groups is given.

The comparison of the rise velocity in Figure 4.17 displays very good agreement. A closer look at the maximum value around $t=0.9$ shows that our numerical result slightly underestimates the peak by $4\cdot10^{-4}$ compared to the other groups. However, note that FreeLIFE overestimates both other benchmark groups in the same range. The maximum rise velocity and the corresponding time are given in Appendix A.3 and A.4.

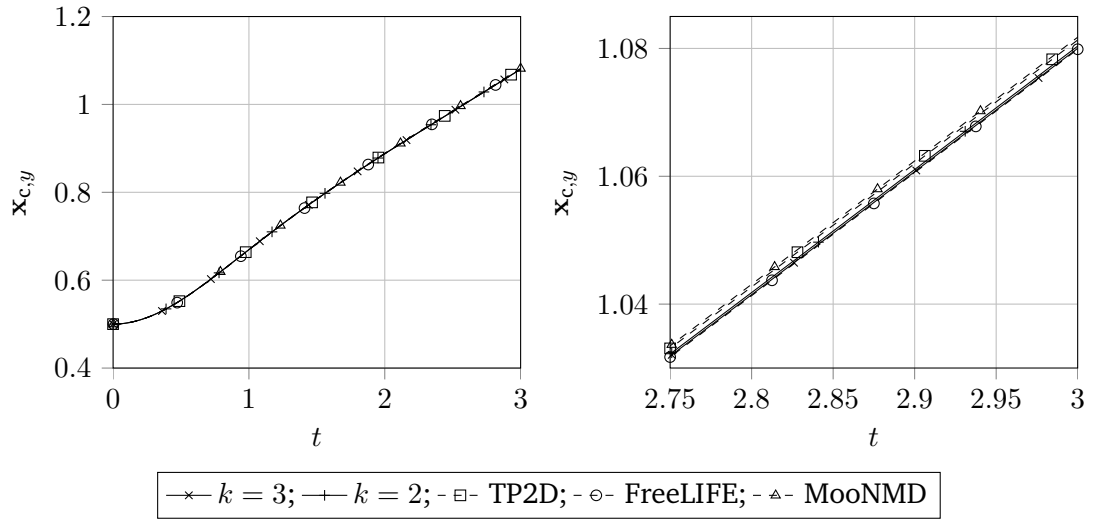


Figure 4.16.: Comparison to benchmark groups for the temporal evolution of the centre of mass (test case 1). The plotted solutions show the respective finest resolution.

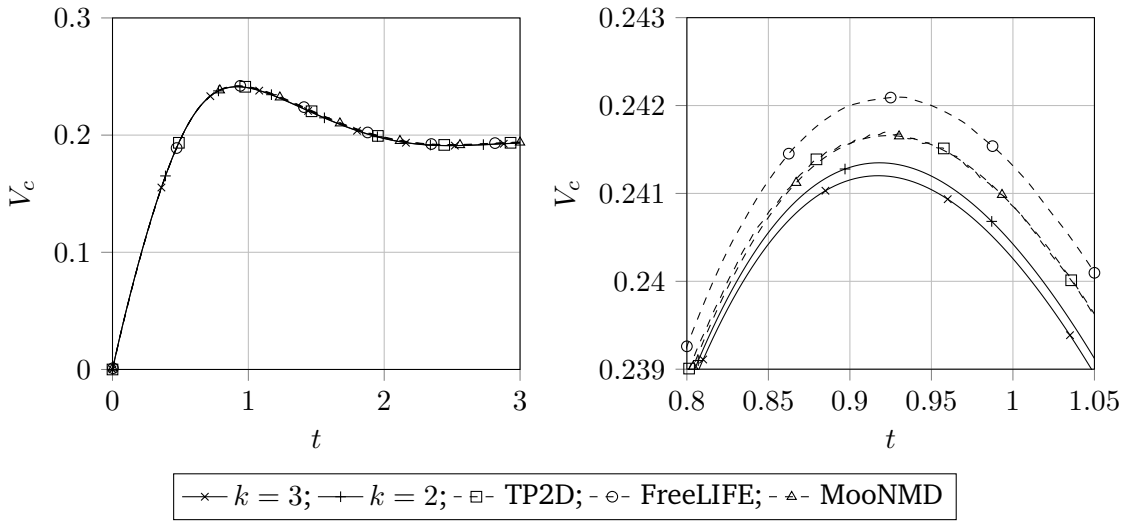


Figure 4.17.: Comparison to benchmark groups for the temporal evolution of the rise velocity (test case 1). The plotted solutions show the respective finest resolution.

Looking at Figure 4.18 the qualitative agreement between the groups is still good for the overall evolution of the circularity, but our numerical result clearly overestimates the minimum and following evolution. Zooming at the time interval $t = [1.8, 2]$ around the minimum, the results of TP2D and MooNMD indistinguishably coincide. The minimum value of FreeLIFE is reached slightly before the other groups. Note that our results are qualitatively closer to the solution of FreeLIFE exhibiting a more gentle slope in the beginning and getting steeper to the end.

A reason for the deviation of the circularity values may be the mass production during the simulation, since the bubble area and the perimeter are directly affected. However, the mass

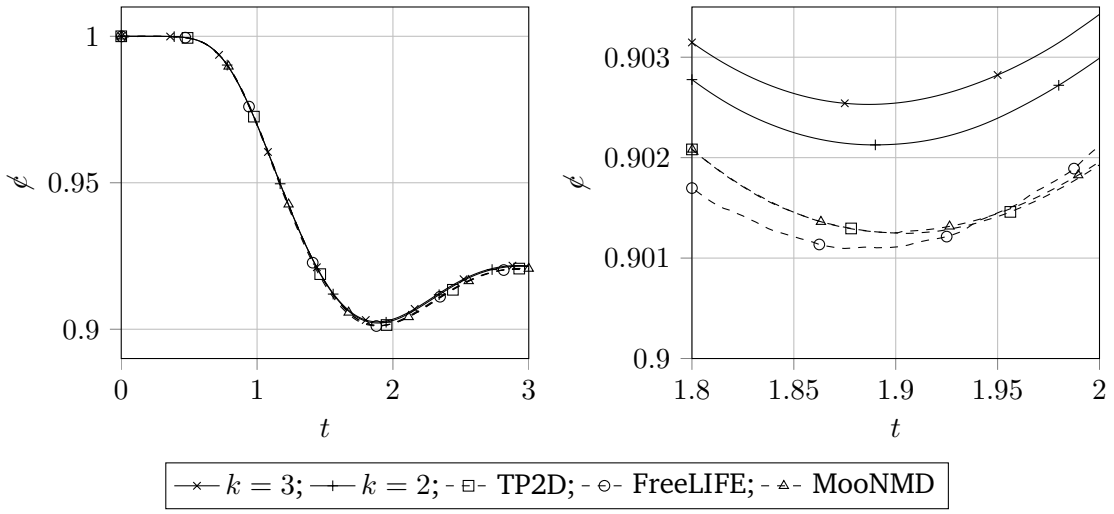


Figure 4.18.: Comparison to benchmark groups for the temporal evolution of the circularity (test case 1). The plotted solutions show the respective finest resolution.

production is comparably small for the finest solutions. Another reason may be the interface evolution algorithm, where the construction of the extension velocity, the actual advection and the projection on a continuous level-set all affects the position of the interface.

4.4.2. Test Case 2

For the second test case we present the finest solutions for $k = 2$ and directly compare them to the solutions of the benchmarks groups. No ROC are compared, but distinct values in the temporal evolution of the benchmark quantities as done in Hysing et al. (2009). The values for the mesh study with $\frac{1}{h} = \{20, 40, 80\}$ are found in Appendix A.5. Due to the stronger deformations during the bubble motion, see Figure 4.19, we perform all runs with an additional adaptive mesh refinement of level 1 at the interface.

In Figure 4.19 the evolution of the bubble shape is shown at the time steps $t = \{0, 1.2, 2.1, 3\}$. Comparing to the first test case, see Figure 4.12, the bubble exhibits a considerably more concave deformation during its rise ($t = 1.2$). This evolves into a cap-like shape of the bubble, where thin filaments starts to emerge from the bubble ($t = 2.1$). These filaments get longer and thinner close to the bubble ($t = 3$). The terminal shape of the bubble assumes the expected dimpled cap.

A comparison between the terminal shapes of the benchmark groups is given in Figure 4.20. The main bubble shape agrees well among all methods only varying in the formation of the filaments. Such filaments are strongly mesh dependent and one cannot expect good agreement in such regions.

However, taking a closer look on the transition between bubble and filaments, two different shapes may be characterized. The filaments are slightly more extended to the outside of the bubble in our solution and the one from MooNMD. On the other side TP2D and FreeLIFE show

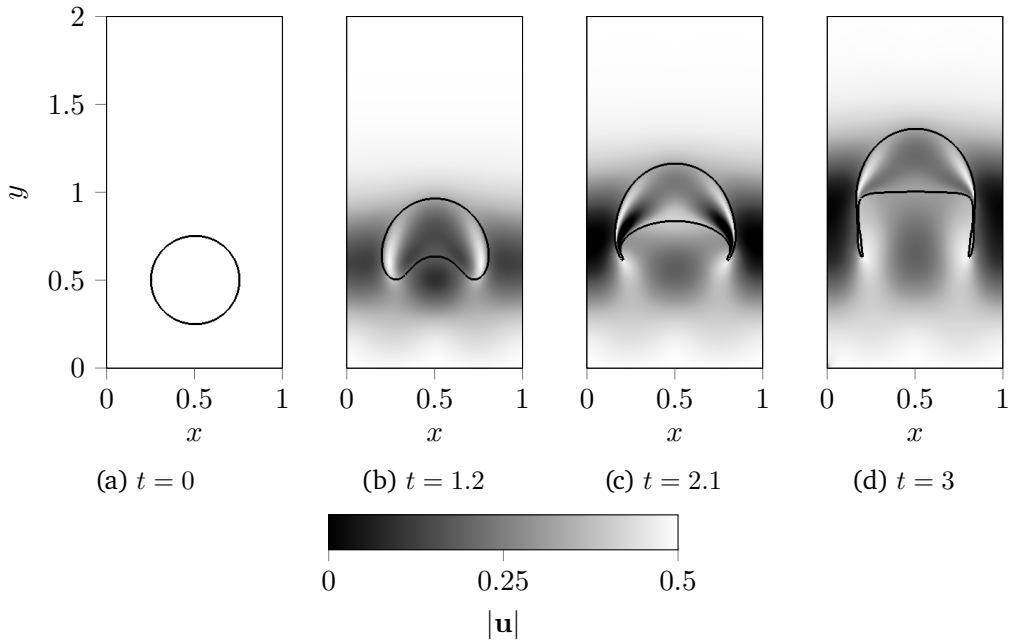


Figure 4.19.: Temporal evolution of the rising bubble benchmark test case 2 for times $t = \{0, 1.2, 2.1, 3\}$. The shown results are computed with a polynomial degree of $k = 2$ on a mesh with $\frac{1}{h} = 80$ and AMR level 1. The plotted field describes the magnitude of the velocity vector field $|\mathbf{u}|$.

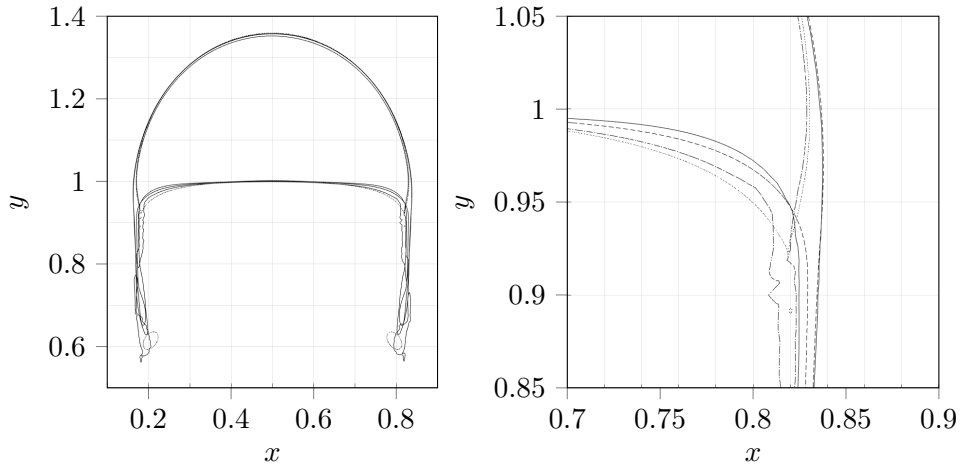


Figure 4.20.: Comparison of the terminal bubble shapes for test case 2: $k = 2$ on mesh $\frac{1}{h} = 80$ with AMR level 1 (solid black), TP2D (dotted), FreeLIFE (dashdotted) and MooNMD (dashed).

are more curved shape at the transition region. The result of TP2D even exhibit a break up of the filaments with two additional satellite droplets.

Comparing the temporal evolution of the benchmark quantities, the agreement of the centre of mass is again very good, see Figure 4.21. However, the deviations to the end of the simulation

is larger compared to the first test case and our numerical result lies in between the benchmark groups. The terminal rise height and other distinct values of the temporal evolution are found in Appendix A.5.

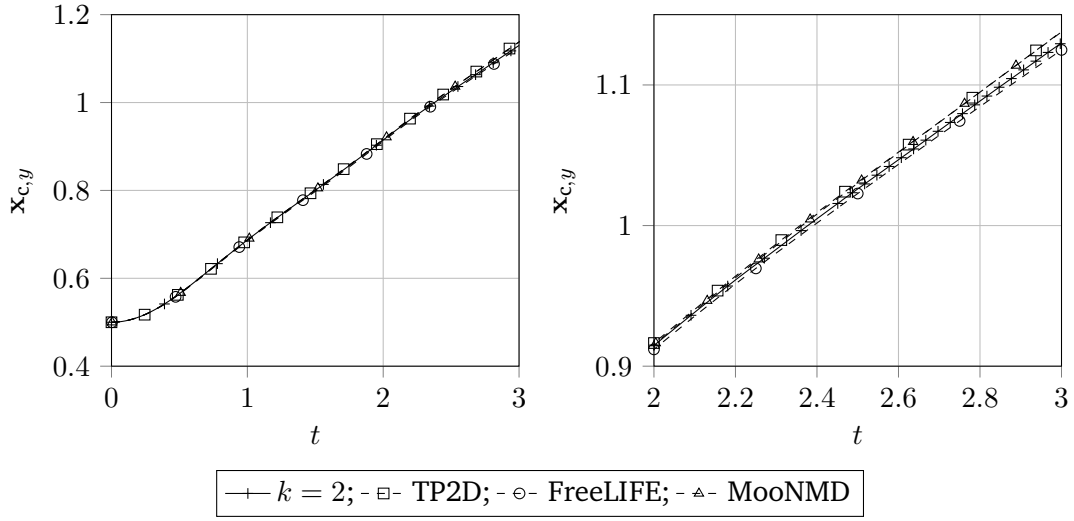


Figure 4.21.: Comparison to benchmark groups for the temporal evolution of the centre of mass (test case 2). The plotted solutions show the respective finest resolution.

Surprisingly, the evolution of the mass centre is not much changed compared to the first test case. However, the rise velocity shows a different characteristic with an additional peak around $t = 2$, which marks the beginning of the emerging filaments from the bubble, see Figure 4.22.

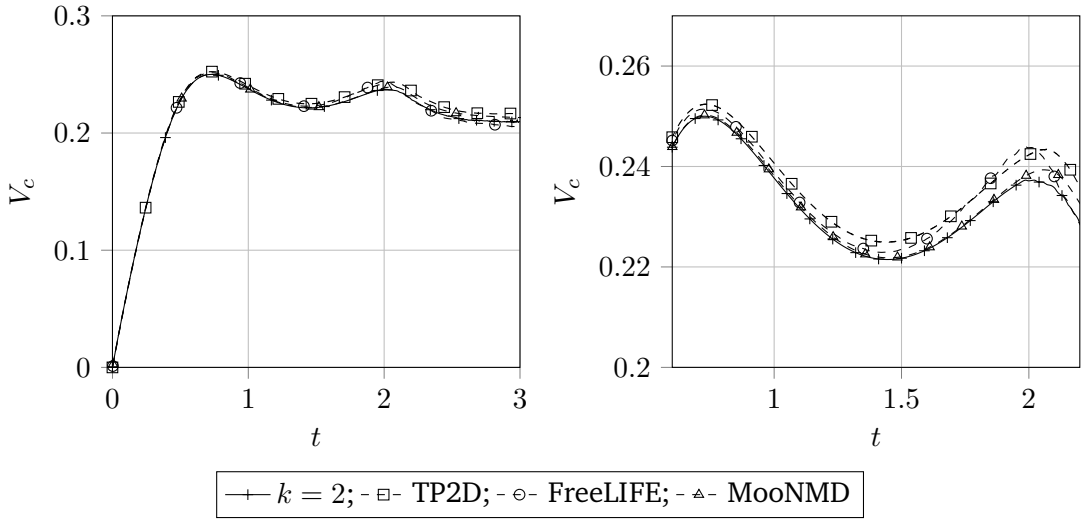


Figure 4.22.: Comparison to benchmark groups for the temporal evolution of the rise velocity (test case 2). The plotted solutions show the respective finest resolution.

The overall agreement on the two-peak characteristic is quite good. The agreement on the first peak is clearly better than on the second peak, which is predicted to be of slightly smaller

magnitude than the first. Taking a closer look, our solution agrees very well with MooNMD underpredicting the whole evolution of rise velocity compared to both other groups. The qualitative agreement between BoSSS and MooNMD also holds for the circularity, see Figure 4.23. In the second test case the circularity does not exhibit a minimum value except for TP2D, which is due to the retraction of the filaments after the break up. The agreement between all methods is very well only up until $t \approx 1.75$.

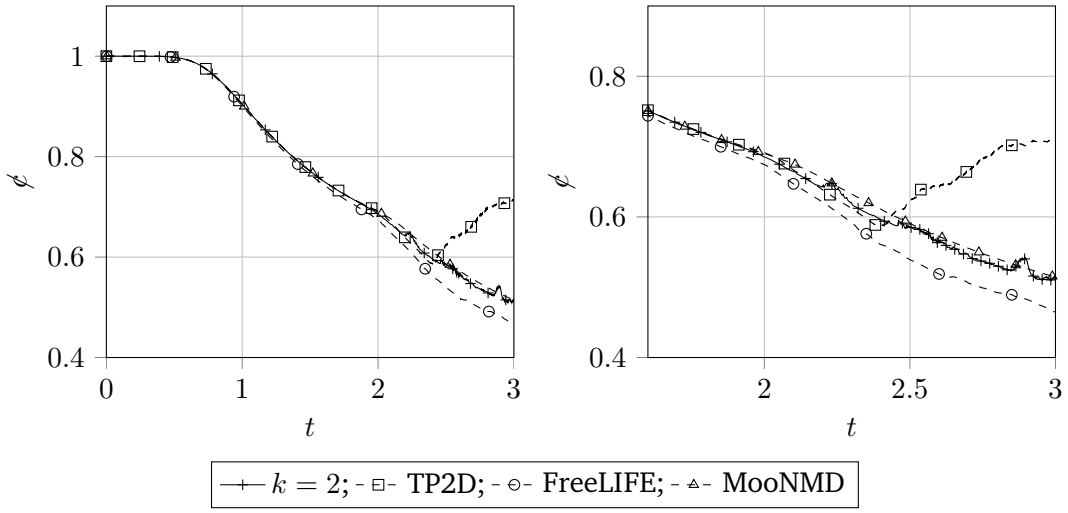


Figure 4.23.: Comparison to benchmark groups for the temporal evolution of the circularity (test case 2). The plotted solutions show the respective finest resolution.

4.5. Conclusion

The developed XNSE-solver shows very good agreement to analytical solutions for surface tension driven capillary waves and benchmark results for the rise of a bubble in a liquid column. For the capillary wave test case a wide range from overdamped to high-oscillatory wave behaviour is presented. For the latter case the numerical results slightly underestimate the damping rate. The agreement is improved by a higher temporal resolution or an increased spatial approximation order. However, higher order convergence rates could not be observed for the amplitude height in this case.

The expected EOC for the bulk properties velocity and pressure ise shown for the artificial steady-state solution of a deformed droplet. A mesh convergence study for polynomial degrees $k = \{2, 3, 4\}$ was performed and a good to very good agreement with the expected convergence order of $\mathcal{O}(h^{k+1})$ is given. Furthermore, the condition number under mesh refinement was investigated, displaying the expected behaviour and thus ensuring a correct and stable implementation. Regarding the stability of the discretization, especially the surface tension force discretization, transient simulations with a circular droplet in equilibrium state were performed. All simulations remained in a stable state during the simulation time, but evaluating the surface divergence indicates some stability issues.

The stability against the capillary time step restriction was investigated for an oscillating droplet. A mesh convergence study was performed with a fixed time steps size, where the time step restriction is already exceeded on the second mesh. However, all simulations showed a stable oscillating behaviour until the end.

The XNSE-solver was tested against other numerical codes within the rising bubble benchmark. Two physical settings were presented exhibiting a stable ellipsoidal bubble shape and a non-convex shape with filaments. The developed solver showed good to very good agreement to the benchmark groups for both settings considering the terminal bubble shape and scalar measure quantities, such as the centre of mass, rise velocity and circularity. The largest deviation from the benchmark groups was recognized for the circularity within the first setting.

Overall, the presented test cases and results attribute the developed solver a validated and reliable prediction of a wide range of typical two-phase flow phenomena. Note that many simulations were performed with an adaptive mesh refinement.

5. Numerical Results of Dynamic Contact Line Flows

In this section the presented problem settings feature contact lines at the boundaries of the computational domain. We consider artificial steady-state simulations of a droplet deposited on a horizontal wall (Section 5.1), and the terminal equilibrium shape of a droplet spreading under the influence of gravity (Section 5.2). The results of the rise of a liquid inside a capillary are presented in Section 5.3.

5.1. Droplet on Slip Wall - Steady-State Simulations

Corresponding to the steady-state simulations presented in Section 4.2.1 we consider a droplet with fixed interface that is deposited in a non-equilibrium state on a slip wall. Again, we are investigating the condition number and the spatial convergence.

The numerical setup is the same as described in Section 4.2.1, but restrict the computational domain to $\Omega = [-1.5, 1.5] \times [0, 1.5]$. Thus, the droplet exhibits a contact angle of $\theta = 90°$. For the static contact angle we consider two cases with $\theta_{\text{stat}} = \{90°, 120°\}$. The entire boundary is imposed with the generalized Navier-slip boundary condition (2.25) and (2.28), where we employ a quasi-static contact angle model with $\beta_L = 0$. For the phase coefficients of friction β_S we also consider two cases with $\beta_S = \{5, 0\}$. The first one describes a slip length of $l_s = 0.1$ in phase $\mathfrak{A}$ (droplet) and $l_s = 0.01$ in phase $\mathfrak{B}$, the second case a free-slip on the entire boundary.

Condition Number Test The condition number study is performed on meshes with 8×16, 16×32, 32×64, 64×128 and 96×192 cells. In Figure 5.1 the results for $\theta_{\text{stat}} = 120°$ are presented. The results for $\theta_{\text{stat}} = 90°$ are found in Appendix A.3. Note that compared to the condition number study in Section 4.2.1, additionally, the stencil condition numbers for the uncut boundary cells are given (pentagon marks). The results of all four cases show the expected behaviour as described before. The slope of the total condition numbers does not exceed the maximum slope of 2 and the maximum stencil condition numbers for all cell types remain nearly unchanged. However, one should observe that the absolute values of the total condition numbers for $\beta_S = 5$ are roughly one order of magnitude larger. The stencil condition numbers on the mesh with 16×32 cells for $\beta_S = 5$ and a static contact angle of $\theta_{\text{stat}} = 120°$ is displayed in Appendix A.4. The distribution and the range of the absolute values are comparable to Figure 4.5.

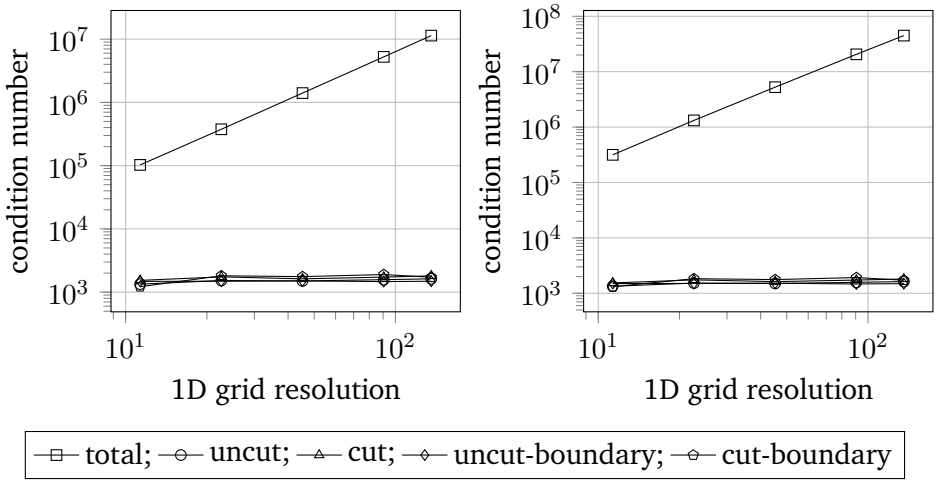


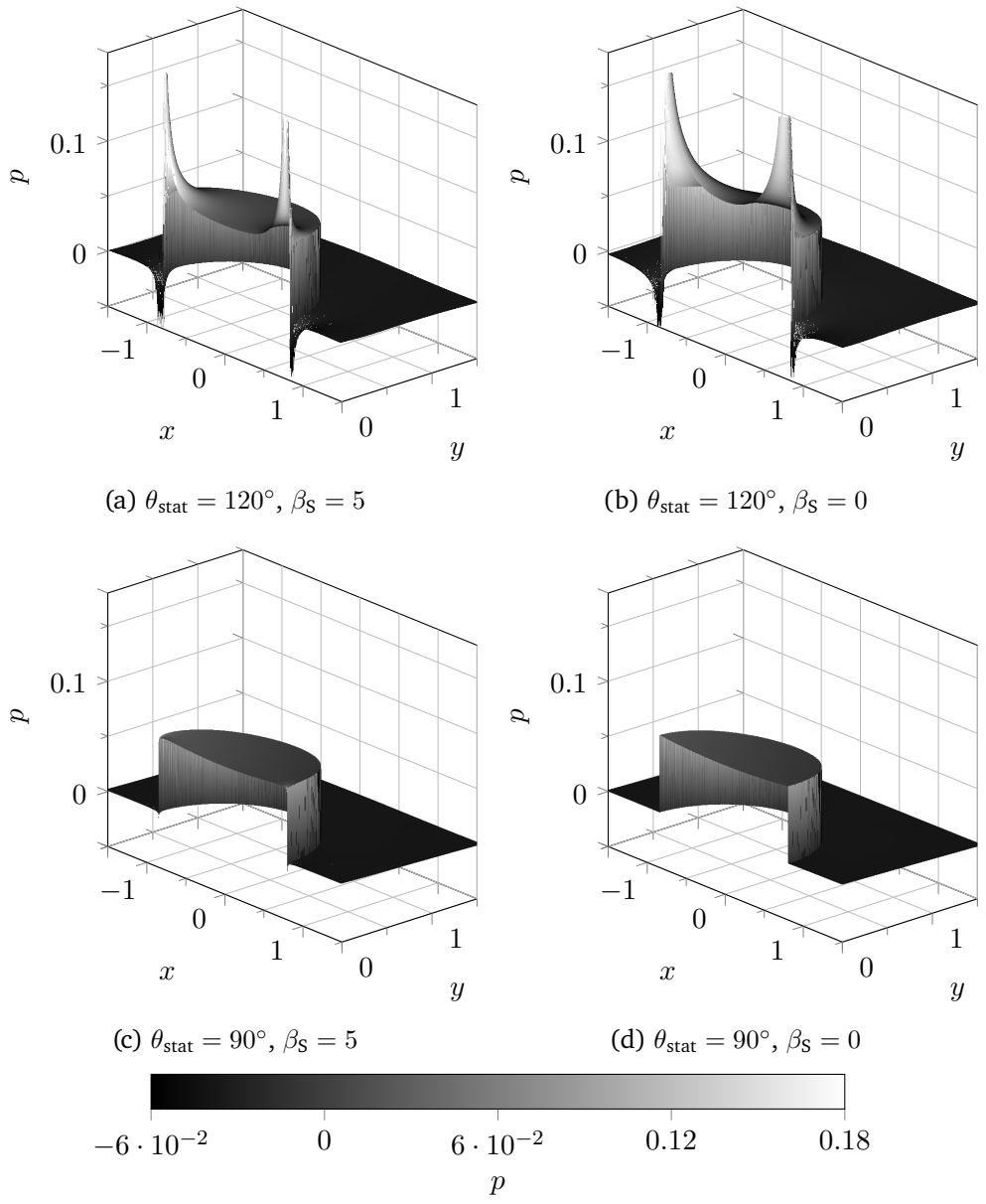
Figure 5.1.: Condition number study for $\theta_{\text{stat}} = 120°$ of a steady-state droplet on a slip wall with $\beta_S = 5$ (left) and $\beta_S = 0$, i.e. free-slip, (right). The study is performed on the following meshes: 8×16, 16×32, 32×64, 64×128, 96×192.

Convergence Study For the convergence study we run simulations for polynomial degrees of $k = \{2, 3, 4\}$ on meshes with 4×8, 8×16, 16×32, 32×64, 64×128, 128×256 and 256×512 cells. The solution on the finest mesh is used as reference solution to compute the L^2-error norms. The EOC for the four considered test cases, i.e $\theta_{\text{stat}} = \{120°, 90°\}$ and $\beta_S = \{5, 0\}$, are given in Table 5.1. The corresponding double-logarithmic plots are given in Appendix A.5 to A.8.

Table 5.1.: EOC for $k = \{2, 3, 4\}$ of the mesh convergence study for a droplet on a slip wall.

$k \backslash k'$	$\theta_{\text{stat}} = 120°$ $\beta_S = 5$	$\theta_{\text{stat}} = 120°$ $\beta_S = 0$	$\theta_{\text{stat}} = 90°$ $\beta_S = 5$	$\theta_{\text{stat}} = 90°$ $\beta_S = 0$
EOC $\|\mathbf{u} - \mathbf{u}_{\text{ref}}\|_2$				
2\1	0.7	1.1	1.8	2.8
3\2	0.9	1.1	1.7	3.8
4\3	1.1	1.1	1.7	4.5
EOC $\|p - p_{\text{ref}}\|_2$				
2\1	0.1	0.2	0.9	1.9
3\2	0.1	0.1	0.8	2.6
4\3	0.1	0.1	0.8	2.8

Looking at the EOC of both cases with $\theta_{\text{stat}} = 120°$ one can see that for all tested polynomial degrees $k = \{2, 3, 4\}$ the convergence order degenerates to EOC ≈ 1 for the velocity norm and, respectively, to EOC ≈ 0.1 for the pressure norm. This is a result of the weak pressure singularity with $p \sim \ln r$ near the contact line, where r denotes the distance to the contact line L (Sprittles and Shikhmurzaev, 2011). Such a solution cannot be approximated accurately by polynomial ansatz functions as used in our discretization. The representation of the pressure singularity on the finest mesh are shown in Figure 5.2 for both cases $\beta_S = \{5, 0\}$ with $\theta_{\text{stat}} = 120°$. The singularity is positive in the droplet phase and negative in the surrounding phase. Note that the pressure singularity inside the droplet is broader for the free-slip case.



(a) $\theta_{\text{stat}} = 120°$, $\beta_{\text{S}} = 5$

(b) $\theta_{\text{stat}} = 120°$, $\beta_{\text{S}} = 0$

(c) $\theta_{\text{stat}} = 90°$, $\beta_{\text{S}} = 5$

(d) $\theta_{\text{stat}} = 90°$, $\beta_{\text{S}} = 0$

Figure 5.2.: Pressure field of a droplet with $\theta_{\text{stat}} = \{120°, 90°\}$ on a slip wall with $\beta_{\text{S}} = 5$ and a free-slip wall with $\beta_{\text{S}} = 0$.

Considering the cases for $\theta_{\text{stat}} = 90°$, where the current contact angle assumes the static equilibrium contact angle, one observes no singularity (Figure 5.2). However, for $\beta_{\text{S}} = 5$ one notices a drop of the pressure level at the contact line. The results of the pressure fields are directly reflected in the corresponding EOC (Table 5.1). For $\beta_{\text{S}} = 5$ the convergence order for both the velocity and pressure norm increases by an order of 1, but still the expected order cannot be recovered. However, looking at the free-slip case we almost regain the expected orders of h^{k+1}. This is in accordance with the findings in Fricke et al. (2019), where it is stated that a regular, non-trivial, quasi-stationary solution only exists in the free-slip case.

5.2. Droplet Spreading under Gravity

For the second test case we consider a semi-circular droplet on the wall with an initial contact angle of $\theta_0 = 90°$ which is not in its equilibrium shape, since the imposed static contact angle is $\theta_{\text{stat}} < 90°$. Thus, the droplet starts to spread in order to regain its equilibrium shape. However, the terminal shape varies under the influence of an additional gravity force $\mathbf{g} = -g\mathbf{e}_y$. For a study with different gravity forces g we follow the setting proposed in Dupont and Legendre (2010). The computational domain is given by $\Omega = [-3r_0, 3r_0] \times [0, 2r_0]$, where the initial droplet radius is set to $r_0 = 0.001$ m. The physical parameters are given in Table 5.2.

Table 5.2.: Physical parameters for a droplet spreading to equilibrium under gravity.

ρ_{L} in $\frac{\text{kg}}{\text{m}^3}$	ρ_{G} in $\frac{\text{kg}}{\text{m}^3}$	μ_{L} in Pa s	μ_{G} in Pa s	σ in $\frac{N}{m}$
1000	1	0.01	10^{-5}	0.072

Imposing a static contact angle of $\theta_{\text{stat}} = 50°$, the shape of the droplet is only characterized by the Eötvös number Eo which is defined in this test case as

$$\text{Eo} = \frac{\rho_{\text{L}} g r_0^2}{\sigma}, \tag{5.1}$$

where $\rho_{\text{L}} \ll \rho_{\text{G}}$ denotes the density of the liquid droplet and ρ_{G} the density of the surrounding gaseous phase. For $\text{Eo} \ll 1$, the terminal shape is dominated by surface tension effects and the droplet assumes a circular cap with the given static contact angle θ_{stat}. Thus, the droplet thickness, i.e. highest point of the droplet shape, is given by

$$e_0 = r_0(1 - \cos\theta_{\text{stat}})\sqrt{\frac{\pi}{2(\theta_{\text{stat}} - \sin\theta_{\text{stat}}\cos\theta_{\text{stat}})}}. \tag{5.2}$$

For $\text{Eo} \gg 1$, gravity forces dominates the shape and the droplet spreads to a puddle with a thickness of

$$e_\infty = 2\sqrt{\frac{\sigma}{\rho_{\text{L}} g}}\sin\left(\frac{\theta_{\text{stat}}}{2}\right), \tag{5.3}$$

which is directly proportional to the capillary length $l_{\text{cap}} = \sqrt{\frac{\sigma}{\rho_{\text{L}} g}}$. In our study we consider the following range of Eötvös numbers $\text{Eo} = \{0.01, 0.1, 1, 2, 5\}$. The corresponding gravity values are given in Table 5.3.

Table 5.3.: Eötvös number study for a speading droplet under gravity.

Eo	0.01	0.1	0.5	1	2	5
g	0.0072	0.072	0.36	0.72	1.44	3.6

The study was performed on a 20×10-mesh with AMR level 1 at the interface and the slip boundaries. The induced slip length is set to $l_{\text{s}} = \frac{h}{100}$. The simulations are done with a polynomial degree of $k = 2$ and run until $t = 1$ with $\Delta t = 10^{-4}$. In Figure 5.3 the normalized thickness $e^* = \frac{e}{e_0}$ of our results are displayed and compared to the results of Dupont and Legendre (2010) and Gründing (2020). The latter presents an exact solution for interface

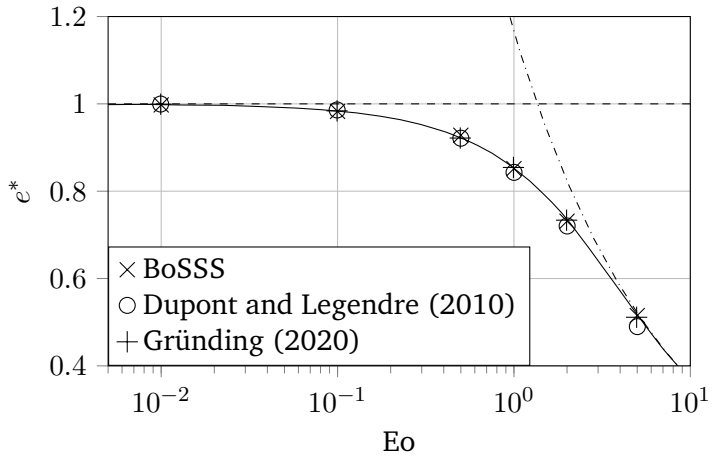


Figure 5.3.: Normalized thickness $e^* = \frac{e}{e_0}$ for a static contact angel $\theta_{\text{stat}} = 50°$ and different Eötvös numbers Eo $= \{0.01, 0.1, 1, 2, 5\}$. The solid line describes the exact solution by Gründing (2020). The dashed line denotes the asymptotic solution for Eo $\ll 1$ and the dash dotted one for Eo $\gg 1$.

shape of the considered inverse problem. In addition to the exact solution, both asymptotic solutions for Eo $\ll 1$, i.e. $e^* = 1$, and Eo $\gg 1$ (Equation (5.3)) are plotted.

The results of BoSSS and Gründing (2020) show excellent agreement to the exact solution for both asymptotic regimes and in the transition region. The results of Dupont and Legendre (2010) also agrees very good with the exact solution but start to slightly underestimate the droplet thickness for higher Eo.

The terminal shapes for Eo $= \{0.01, 2, 5\}$ of our simulations are shown in Figure 5.4. One can clearly see the circular shape for the low Eötvös region (Eo $= 0.01$) and the deformation to a puddle for higher Eo. In the transition one notices the receding surface tension forces and increasing effect of gravity. All shapes exhibit the imposed static contact angle of $\theta_{\text{stat}} = 50°$.

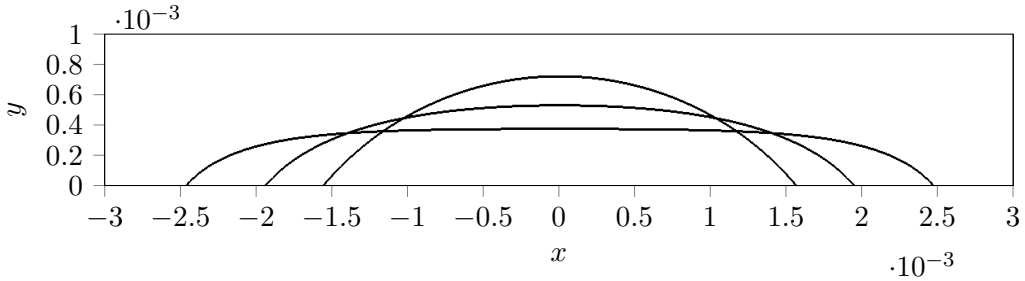


Figure 5.4.: Terminal shapes of a spreading droplet under gravity for Eo $= \{0.01, 2, 5\}$. The semi-circular shape corresponds to Eo $= 0.01$ and the low puddle to Eo $= 5$

5.3. Capillary Rise

In this section we present the results of a benchmark study regarding the rise of a liquid within a capillary. The presented results are already published in Gründing et al. (2020), where the author of this work contributes as one of the four benchmark groups.

The benchmark setting considers the case of a liquid phase $\mathfrak{A}$ rising between two planar surfaces with a gap width of $2r_{\text{cap}}$, see Figure 5.5. The left and right boundary are imposed with the Navier-slip boundary condition (2.25), where a static contact angle $\theta = \theta_{\text{stat}}$ is enforced, i.e. $\beta_L = 0$ in context of the generalized Navier-slip boundary condition (2.28). At the lower and upper boundary $\partial\Omega_{\text{outlet}}$ we impose

$$p = 0 \quad \text{and} \quad \partial_{\mathbf{n}_{\partial\Omega}} \mathbf{u}_{\mathbf{n}_{\partial\Omega}}, \tag{5.4}$$

where $\mathbf{u}_{\mathbf{n}_{\partial\Omega}}$ denotes the normal component of the velocity w.r.t. $\partial\Omega_{\text{outlet}}$. Since the lower and upper pressure are set equal, the hydrostatic pressure of the gaseous phase $\mathfrak{B}$ is neglected. The gravity force is given by $\mathbf{g} = -g\mathbf{e}_y$. Note that the test case exhibits a symmetry w.r.t. the y-axis, thus allowing the computation of only one half and imposing a free-slip boundary condition at $x = 0$ with $\theta = 90°$.

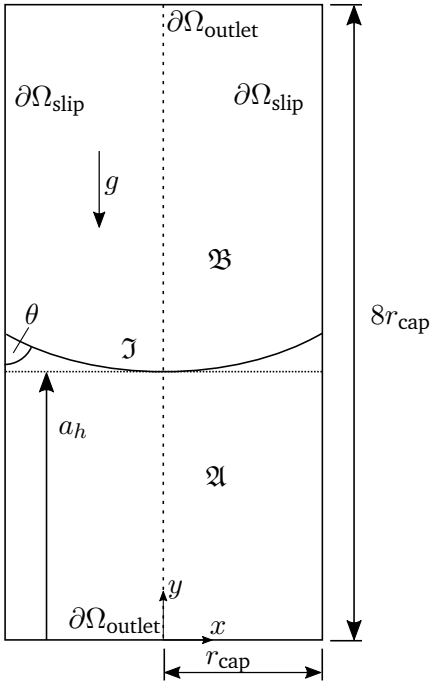


Figure 5.5.: Setup of the capillary rise benchmark

For the choice of the geometric and physical parameters the following requirements apply according to Gründing et al. (2020), where a detailed derivation is given:

- The interface has to maintain a circular shape during the rise in the capillary.
- The setup must yield a rise height that differs significantly from its initial height in order to form various dynamical behaviours.

- The simulation domain should be small enough to reduce the computational costs.
- The physical parameters should be chosen such that the necessary number of time steps is as small as possible.
- The influence of the gaseous phase should be small to allow for comparisons between single-phase free surface models and two-phase models.

The half gap width is set to $r_{\text{cap}} = 5 \cdot 10^{-3}$ m and the height of the computational domain to $8r_{\text{cap}}$. The static contact angle at the slip walls $\partial\Omega_{\text{slip}}$ is fixed to $\theta_{\text{stat}} = 30°$, ensuring that r_{cap} is small compared to the capillary length l_{cap}, i.e. Eo $\ll 1$. Following the dimensional analysis of the classical reduced model (Lucas, 1918; Washburn, 1921; Rideal, 1922) provided by Fries and Dreyer (2008) the considered problem is subject to one single non-dimensional group Π that for the 2D case reads

$$\Pi = \sqrt{\frac{9\sigma \cos\theta \mu^2}{\rho^3 g^2 r^5}}. \tag{5.5}$$

In order to cover a wide range of different physical behaviours we investigate the range of $\Pi = \{0.1, 0.5, 1, 10, 100\}$. Thus, we consider regimes ranging from highly oscillatory (small Π) to a monotonic rise (large Π). The corresponding physical parameters of the liquid phase $\mathfrak{A}$ are given in Table 5.4. Ensuring a comparable small effect of the gaseous phase $\mathfrak{B}$, the density and viscosity ratio are set to $\frac{\rho_{\mathfrak{A}}}{\rho_{\mathfrak{B}}} = 1000$ and $\frac{\mu_{\mathfrak{A}}}{\mu_{\mathfrak{B}}} = 1000$.

Table 5.4.: Physical parameters and simulation times for the capillary rise benchmark study.

Π	$\rho_{\mathfrak{A}}$ in $\frac{\text{kg}}{\text{m}^3}$	$\mu_{\mathfrak{A}}$ in Pa s	g in $\frac{\text{m}}{s^2}$	σ in $\frac{N}{m}$	t_{end} in s
0.1	1663.8	0.01	1.04	0.2	0.69
0.5	133.0	0.01	6.51	0.1	14.0
1	83.1	0.01	4.17	0.04	1.1
10	3.3255	0.01	26.042	0.01	2.7
100	0.33255	0.01	26.042	0.001	27.0

Apex Height and Initial Condition The quantity that is tracked during the simulations and is compared to is the evolution of the interface apex height a_h. This height denotes the lowest point on the circular interface $\mathfrak{I}$ (Figure 5.5). In order to set the initial apex height and the circular shape of the interface a horizontal interface is primarily initialized at $a_{h,0} = 2r_{\text{cap}}$ and the lower boundary condition is set to a no-slip condition. Then the simulation is run until the steady equilibrium state, i.e circular shape with imposed static contact angle θ_{stat}, is obtained. For the benchmark study this start-up time needs to be at least 0.1 s. After this initialization the simulation is restarted with the outlet boundary condition (5.4) and $\mathbf{u} = 0$ as the initial condition.

We want to remark that the stationary apex height $a_{h,\infty}$ for this benchmark study is the same for all Π and is given as follows

$$a_{h,\infty} = a_{h,\text{Jurin2D}} - \hat{a}_h, \quad \hat{a}_h = \frac{r_0}{2\cos\theta_{\text{stat}}}\left(2 - \sin\theta_{\text{stat}} - \frac{\arcsin\left(\cos\theta_{\text{stat}}\right)}{\cos\theta_{\text{stat}}}\right). \tag{5.6}$$

Here, $a_{h,\text{Jurin2D}}$ denotes Jurin's height (Jurin, 1719) adapted for the 2D case with

$$a_{h,\text{Jurin2D}} = \frac{\sigma \cos \theta_{\text{stat}}}{r_0 \rho g}, \tag{5.7}$$

which is a good estimate for the stationary rise height, but neglects the liquid area between the apex height and the interface meniscus, see Figure 5.5. Therefore, a correction term $\hat{a}_h$ is added that accounts for this missing area of liquid.

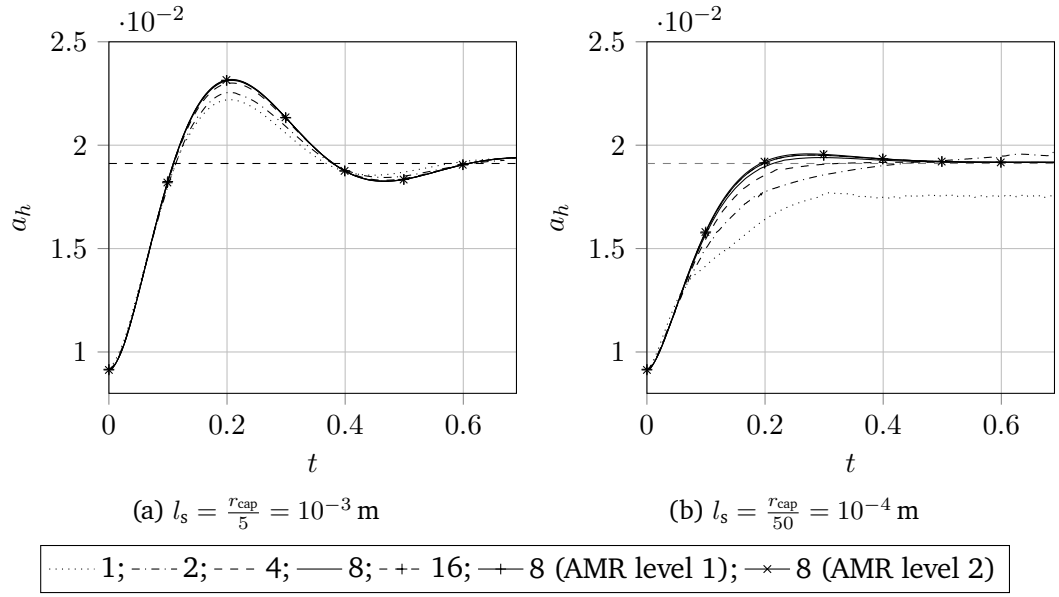
Benchmark Groups for Comparison This benchmark for a capillarity-dominated wetting process was developed within the CRC 1194, see Section 1. Besides our group three other groups, also part of the CRC, were contributing to this benchmark study. These are:

- an Arbitrary Lagrangian-Eulerian (ALE) approach (OpenFOAM solver interTrackFoam),
- a geometric volume of fluid (VOF) code (FS3D, Free Surface 3D),
- an algebraic VOF method (OpenFOAM solver interFoam).

One major difference between all the groups is the numerical representation of the interface. In the ALE approach a part of the numerical mesh represents the interface and, consequently, when the interface moves the interface mesh has to follow the deformation. The remaining part of the mesh follows this deformation in order to maintain a sufficient mesh quality. Note that the ALE method only considers the liquid phase. The VOF methods describe the interface by an indicator function, where in each cells intersected by the interface a value between 0 and 1 indicates the fill level of one phase. Such an interface representation leads to algebraic VOF methods, whereas geometric VOF methods reconstruct the interface location inside each cell, e.g. piece-wise linear. For the latter the effect of numerical dissipation is reduced.

5.3.1. Convergence Study

Before we discuss the Π-study, a convergence study for $\Pi = 1$ with two different slip lengths $l_\text{s} = \left\{\frac{r_\text{cap}}{5}, \frac{r_\text{cap}}{50}\right\}$ is considered. For our study we run the simulations on five meshes with $\frac{r_\text{cap}}{h} = [1, 2, 4, 8, 16]$. Furthermore, we run two simulations on $\frac{r_\text{cap}}{h} = 8$ with additional AMR (constantInterface + navierSlipRefined) with level 1 and 2. Note that we exploit the axial symmetry of the problem, thus, the computational domain is set to $\Omega = [0, r_\text{cap}] \times [0, 8 r_\text{cap}]$. For all the following simulations we use a polynomial degree of $k = 2$ for the velcocities and $k - 1$ for the pressure. A BDF2 scheme is applied for the temporal discretization. The time step size for the convergence study is fixed for all runs and chosen according to Equation (3.119). The results for both slip lengths is displayed in Figure 5.6, where additionally the corrected stationary rise height $a_{h,\infty}$ (5.6) is given. For both slip lengths we observe a very good convergence for the finest resolved solutions. A remarkable observation is that the qualitative agreement for the coarsest solution (one single cell per half gap width) of $l_\text{s} = \frac{r_\text{cap}}{5}$ is still very good. For $l_\text{s} = \frac{r_\text{cap}}{5}$ the coarsest solution clearly underpredicts the stationary rise height. It is surprising that the solutions for the reduced slip length $l_\text{s} = \frac{r_\text{cap}}{50} = 10^{-4}\,\text{m}$, which is still far away from a nano-scale slip, barely show any oscillations. However, the stationary rise height agrees very well with the corrected rise height $a_{h,\infty}$ (dashed line). One should



(a) $l_\mathrm{s} = \frac{r_\mathrm{cap}}{5} = 10^{-3}\,\mathrm{m}$ (b) $l_\mathrm{s} = \frac{r_\mathrm{cap}}{50} = 10^{-4}\,\mathrm{m}$

Figure 5.6.: Convergence study of the capillary rise with $\Pi = 1$ for two different slip lengths $l_\mathrm{s} = \left\{\frac{r_\mathrm{cap}}{5}, \frac{r_\mathrm{cap}}{50}\right\}$.

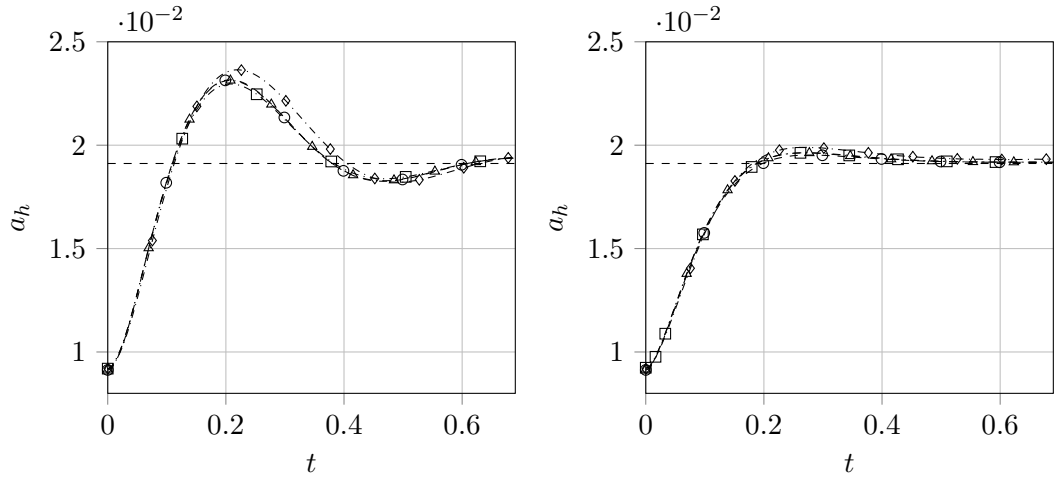
note that such a behaviour is not captured by the classical model (Lucas, 1918; Washburn, 1921; Rideal, 1922): A qualitative change from the oscillatory to the overdamped regime due to a change of the slip length.

Comparison to Benchmark Groups In Figure 5.7 the finest solutions of each benchmark group is plotted. The qualitative agreement between all groups is quite good, where our numerical results show excellent agreement with FS3D. The oscillation amplitude of interTrackFoam is slightly smaller for $l_\mathrm{s} = \frac{r_\mathrm{cap}}{5}$. In comparison, the solutions computed with interFoam oscillate with a slightly stronger amplitude as well as with a shorter frequency in both cases.

The corresponding convergence behaviour for each method is given in Figure 5.8. The normalized maximum difference $\Delta a^*_{h,\max}$ to the finest solution is plotted over the number of cells per half gap width. The maximum difference $\Delta a_{h,\max}$ is given by $\Delta a_\mathrm{max} = \max\left(a_{h,n}(t) - a_{h,N}(t)\right)$, where $a_{h,n}(t)$ denotes the solution for n cells per half gap width and $a_{h,N}(t)$ the corresponding finest solution. The difference is normalized using the corrected stationary rise height (5.6). Note that the XDG-method for both slip lengths allows a comparably smaller number of cells for a fixed accuracy. This is especially pronounced for the smaller slip length, which may be explained by the better approximation of the interface with a second order polynomial.

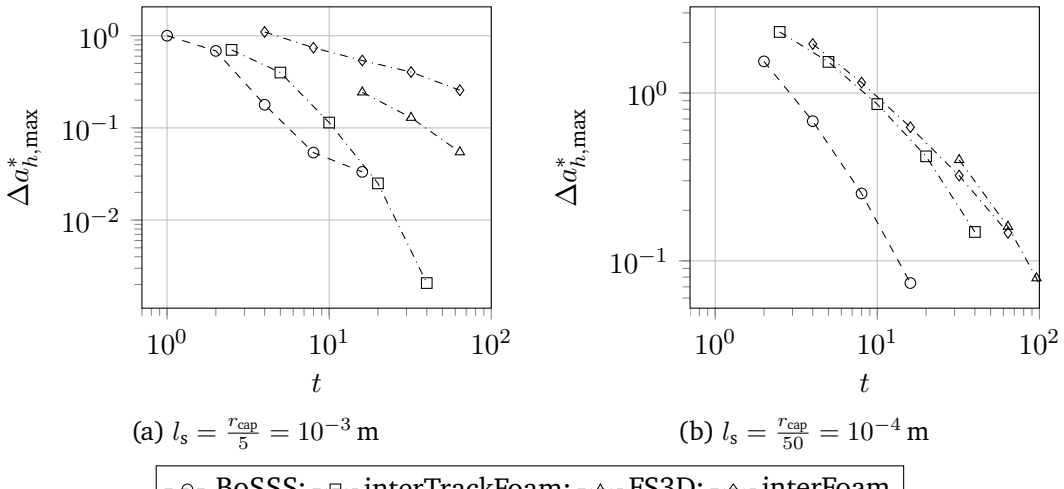
5.3.2. Π-Study

The Π-study is performed with a slip length of $l_\mathrm{s} = \frac{r_\mathrm{cap}}{5}$. Our solutions are computed on a mesh with $\frac{r_\mathrm{cap}}{h} = 8$ and the time step size for each Π is set to the corresponding capillary time step restriction (3.119). Note that for $\Pi = 0.1$ we additionally used AMR on level 1 and applied a reinitialization every 250 time steps. In Figure 5.9 to Figure 5.12 the results of all



(a) $l_s = \frac{r_{cap}}{5} = 10^{-3}\,\mathrm{m}$ (b) $l_s = \frac{r_{cap}}{50} = 10^{-4}\,\mathrm{m}$

Figure 5.7.: Comparison of the benchmark groups' finest solution of the convergence study for the capillary rise ($\Pi = 1$) for two different slip lengths $l_s = \left\{\frac{r_{cap}}{5}, \frac{r_{cap}}{50}\right\}$.



(a) $l_s = \frac{r_{cap}}{5} = 10^{-3}\,\mathrm{m}$ (b) $l_s = \frac{r_{cap}}{50} = 10^{-4}\,\mathrm{m}$

Figure 5.8.: Comparison of the normalized maximum difference $\Delta a^*_{h,\max}$ for the convergence study of the capillary rise ($\Pi = 1$) for two different slip lengths $l_s = \left\{\frac{r_{cap}}{5}, \frac{r_{cap}}{50}\right\}$.

benchmark groups for $\Pi = \{0.1, 0.5, 10, 100\}$ are presented. The results of FS3D describe the finest solution of a convergence study, which is done for each Π, and thus serves as a converged reference solution. Furthermore, the corrected rise height (5.6) as a dashed line is given as a reference.

Throughout the study all benchmark groups show good to excellent qualitative agreement to each other. For small Π all methods predict strong oscillations which are damped over time and level at the corrected rise height $a_{h,\infty}$. For $\Pi = 0.1$ the solutions coincide up to

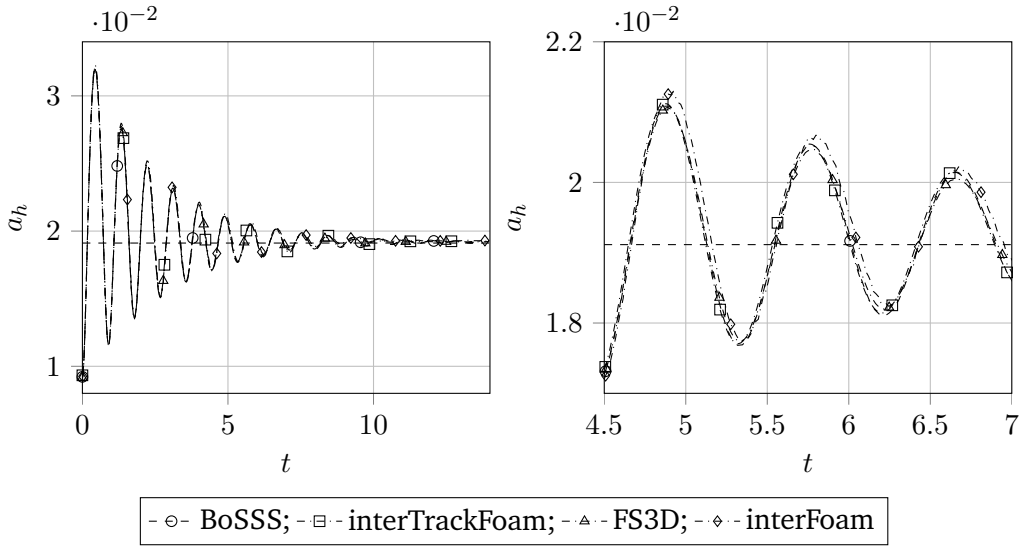


Figure 5.9.: Comparison of the benchmark groups for the capillary rise of $\Pi = 0.1$. On the right hand side a zoom for $t = [4.5, 7]$ is given.

approximately $t = 5\,\mathrm{s}$ (Figure 5.9 on the right). From then on, first interFoam, followed by FS3D, start to shorten the oscillation frequency compared to the remaining solutions. Furthermore, the amplitude height reduces for FS3D, whereas interFoam exhibits for all lower Π an overestimation of the amplitude height. With increasing Π the numerical solutions show less oscillations.

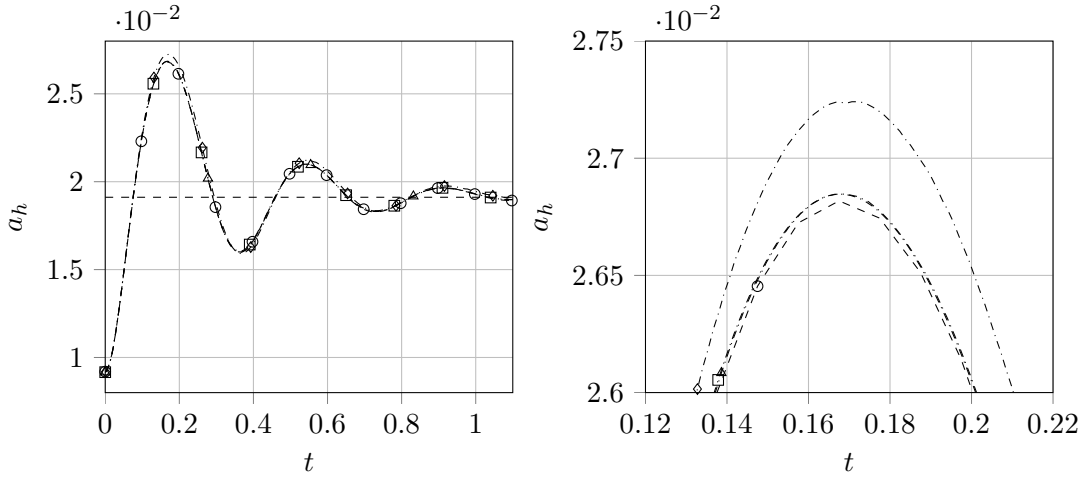


Figure 5.10.: Comparison of the benchmark groups for the capillary rise of $\Pi = 0.5$. On the right hand side a zoom on the first peak for $t = [0.12, 0.22]$ is given.

The cases for $\Pi = 10, 100$ (Figure 5.11 and Figure 5.12) do not exhibit any oscillations. All methods predict the stationary rise height $a_{h,\infty}$ in good agreement with the corrected rise height (5.6). Overall, the dynamic behaviour of the numerical solutions present a stronger

dependence on Π for decreasing values of Π.

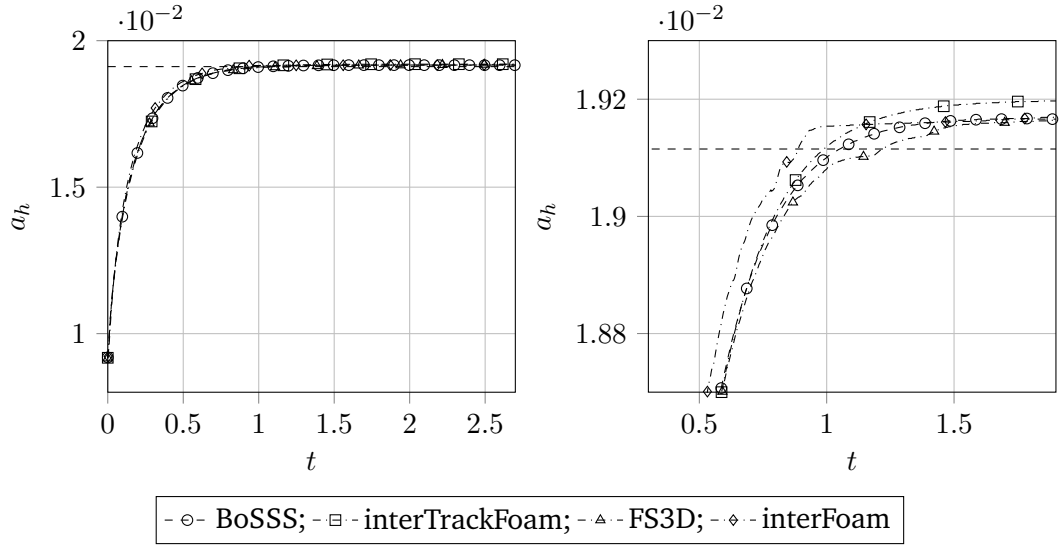


Figure 5.11.: Comparison of the benchmark groups for the capillary rise of $\Pi = 10$. On the right hand side a zoom for $t = [0.3, 1.9]$ is given.

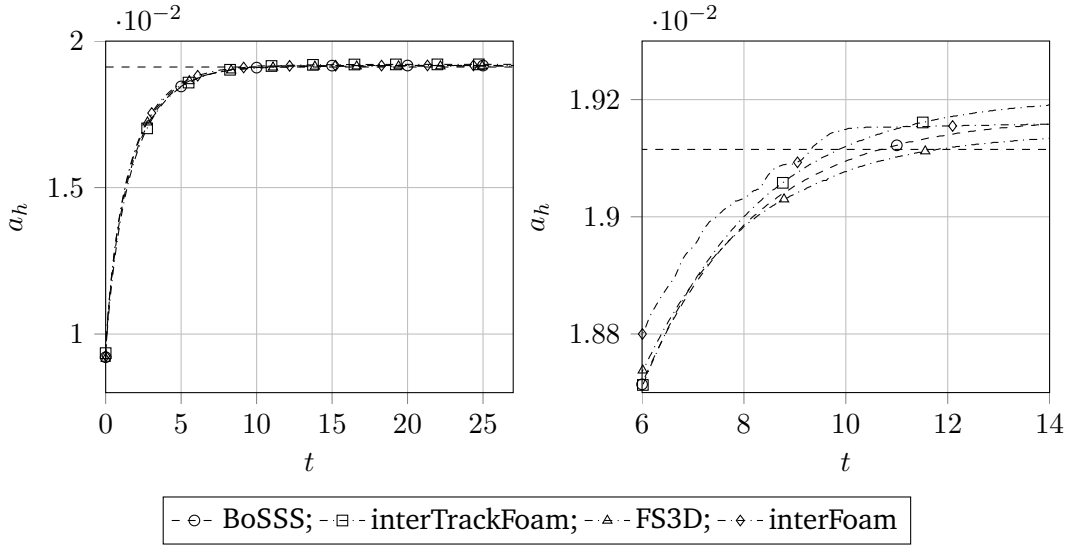


Figure 5.12.: Comparison of the benchmark groups for the capillary rise of $\Pi = 100$. On the right hand side a zoom for $t = [6, 14]$ is given.

5.4. Conclusion

In this chapter the verification and validation of the XNSE-solver was extended to flow problems including dynamic contact lines at the boundaries. The boundaries in this case were imposed by the generalized Navier boundary condition. The condition number study for a fixed droplet

on a slip wall ensures a correct and stable implementation of the GNBC in the solver framework. This also holds for the free-slip boundary condition. The corresponding convergence study showed two characteristics. First, the high-order spatial convergence rate degenerates due to the inherent weak pressure singularity near the contact line. Second, the high-order convergence is recovered for the quasi-stationary case if a free-slip boundary condition is imposed. This marks the only regular non-trivial solution for contact line problems with slip boundary conditions (Fricke et al., 2019). Thus, the ability of DG methods to identify the well-posedness of a discretized problem was demonstrated.

The results for a spreading droplet under the influence of gravity forces showed excellent agreement to the exact solution for the terminal droplet height. Both regimes from a circular shape dominated by surface tension effects to a puddle dominated by gravity forces were presented.

The studies of a new benchmark setup for the capillary rise showed good to very good agreement between all participating benchmark groups. A convergence study for the capillary rise height was performed for two different slip lengths. All groups predict a qualitative change from a oscillatory to a overdamped regime by reducing the slip length. Such a behaviour is not captured by the classical model for the rise height. Furthermore, the groups were compared for different physical behaviours ranging from highly oscillatory to monotonic rise. The presented results are published in Gründing et al. (2020).

6. The Multi-phase Flow Solver Coupled with the Heat Equation

In this chapter the extension of the multiphase flow solver to problems exhibiting evaporation at the interface is presented. Therefore, the discretization of the two-phase Navier-Stokes equations is adapted to allow an additional mass flux across the interface (Section 6.1.1). In Section 6.1.2 an XDG discretization for the two-phase heat equation with energy flux across the interface is proposed. The coupling of the heat equation with the XNSE-solver and the computation of the evaporative mass is described in Section 6.2.

There is little work done for the considered coupled problem in the context of extended numerical methods. In Martin et al. (2018) a model for solidification problems including convection is presented for the coupling of the Stefan problem with the Stokes problem. The XFEM is used to discretize the flow properties as well as the temperature field. A phase field formulation of the Navier-Stokes equations is considered in Zabaras et al. (2006). However, there are many works thath consider the XFEM for the Stefan problem, e.g. (Chessa et al., 2002; Merle and Dolbow, 2002; Bernauer and Herzog, 2012; Martin et al., 2016). In the context of a generalized multiscale DG method Stepanov et al. (2018) provides a numerical solution of a heat transfer problem with phase change in heterogeneous domains. The IP method is used for the global coupling on a coarse grid.

6.1. The Spatial Discretization for Evaporation Problems

In this section we formulate the spatial discretization for two-phase flows with evaporation at the interface in context of the XDG method. This includes an extension to the Navier-Stokes discretization (3.38) and the introduction of the two-phase heat equation. A fixed interface at $\mathfrak{I} = \mathfrak{I}(t^{n+1})$ is considered and the integration domains are set to the time-level t^{n+1}.

6.1.1. Two-Phase Navier-Stokes Equations with Mass Flux at the Interface

For the extension of the transient two-phase incompressible Navier-Stokes equations (2.3) with an additional mass flux $\dot{m}$ across the interface, see Equation (2.30) and Equation (2.33), we propose the following discretization according to Schott (2017): Find $(\mathbf{u}^{n+1}, p^{n+1}) \in \mathbb{V}^{\mathrm{X}}_{\mathbf{k}}$, such that $\forall (\mathbf{v}, q) \in \mathbb{V}^{\mathrm{X}}_{\mathbf{k}}$

$$m(\partial_t \mathbf{u}|_{t^{n+1}}, \mathbf{v}) + c(\mathbf{u}^{n+1}, \mathbf{u}^{n+1}, \mathbf{v}) + b(p^{n+1}, \mathbf{v}) - a(\mathbf{u}^{n+1}, \mathbf{v}) - b(q, \mathbf{u}^{n+1}) = g(\mathbf{v}, q) + f(\mathbf{v}, q). \tag{6.1}$$

All linear forms except $f(\mathbf{v}, q)$ are known as given in the variational formulation (3.38) of the two-phase incompressible Navier-Stokes equation without mass flux. Note that the temporal discretization is unchanged. In $f(\mathbf{v}, q)$ consistently added interface coupling terms are summarized

$$f(\mathbf{v}, q) = f_c(\mathbf{v}) + f_a(\mathbf{v}) + f_b(q). \tag{6.2}$$

The first term $f_c(\mathbf{v})$ enforces the additional convective contribution of the mass flux in the extended dynamic jump condition (2.32)

$$f_c(\mathbf{v}) = \oint_{\mathfrak{I}} \dot{m}^2 \left[\!\left[\rho^{-1}\right]\!\right] \mathbf{n}_{\mathfrak{I}} \cdot \{\{\mathbf{v}\}\} \ \mathrm{d}S. \tag{6.3}$$

The second term $f_a(\mathbf{v})$ incorporates the velocity jump (2.30) for the viscous terms discretized by the extended SIP method

$$\begin{aligned} f_a(\mathbf{v}) = &- \oint_{\mathfrak{I}} \left\{\left\{\mu \left(\nabla_h \mathbf{v} + \nabla_h \mathbf{v}^T\right)\right\}\right\} \mathbf{n}_{\mathfrak{I}} \cdot \left[\!\left[\rho^{-1}\right]\!\right] \dot{m} \ \mathrm{d}S \\ &+ \oint_{\mathfrak{I}} \eta \dot{m} \left[\!\left[\rho^{-1}\right]\!\right] \cdot [\![\mathbf{v}]\!] \ \mathrm{d}S. \end{aligned} \tag{6.4}$$

The enforcement of the extended kinematic jump condition (2.30) for the continuity equation is given by

$$f_b(q) = - \oint_{\mathfrak{I}} \dot{m} \left[\!\left[\rho^{-1}\right]\!\right] \{\{q\}\} \ \mathrm{d}S. \tag{6.5}$$

6.1.2. Two-Phase Heat Equation with Mass Flux at the Interface

For the discretization of the two-phase heat equation (2.9), we rewrite the material derivative to its conservative form and we neglect the dissipation term on the right hand side. Thus, we are considering the following heat equation variant

$$\rho c \left(\frac{\partial T}{\partial t} + \nabla \cdot (\mathbf{u} T) \right) = k \Delta T. \tag{6.6}$$

Assuming that $T = T_{\text{sat}}$ on the interface $\mathfrak{I}$, we propose the following discretization for the heat equation (6.6) with the Dirichlet and Neumann boundary conditions (2.10): Find $T^{n+1} \in \mathbb{P}_k^{\mathrm{X}}(\mathfrak{K}_h, t^{n+1})$, such that $\forall r \in \mathbb{P}_k^{\mathrm{X}}(\mathfrak{K}_h, t^{n+1})$

$$m_H(\partial_t T|_{t^{n+1}}, r) + c_H(\mathbf{u}^{n+1}; T^{n+1}, r) - a_H(T^{n+1}, r) = t_H(r) + s_H(r). \tag{6.7}$$

The first bilinear form on the left hand side corresponds to the time derivative term $\partial_t T$ and is defined as

$$m_H(T, r) = \int_{\Omega} \rho c T \cdot r \ \mathrm{d}V. \tag{6.8}$$

The bilinear form $c_H(\mathbf{u}^*; T, r)$ with the flow velocity $\mathbf{u}^*$ as parameter describes the discretization of the convective term, where a local Lax-Friedrichs flux in scalar formulation is employed

$$\begin{aligned} c_H(\mathbf{u}^*; T, r) = &- \int_{\Omega} \rho c T \mathbf{u}^* \cdot \nabla_h r \ \mathrm{d}V \\ &- \oint_{\Gamma_{\text{int}} \cup \Gamma_{\text{N}}} \left(\{\{T \mathbf{u}^*\}\} \cdot \mathbf{n}_{\mathfrak{I},\Gamma} + \frac{\lambda}{2} [\![T]\!] \right) \cdot [\![\rho c r]\!] \ \mathrm{d}S + c_{H,\mathfrak{I}}(\mathbf{u}^*; T, r). \end{aligned} \tag{6.9}$$

For the enforcement of $T = T_{\text{sat}}$ on the interface $\mathfrak{I}$ the flux $c_{H,\mathfrak{I}}(-;-,-)$ is given by

$$c_{H,\mathfrak{I}}(\mathbf{u}^*;T,r) = -\oint_{\mathfrak{I}} [\![\rho c(\mathbf{u}^* \cdot \mathbf{n}_{\mathfrak{I}})r]\!]\, T_{\text{sat}} + \{\{\rho c \lambda T r\}\} - \{\{\rho c \lambda r\}\}\, T_{\text{sat}}\, \mathrm{d}S. \tag{6.10}$$

The conductive terms are discretized using the standard SIP method

$$\begin{aligned} a_H(T,r) = &-\int_{\Omega} k\nabla_h T \cdot \nabla_h r \, \mathrm{d}V \\ &+ \oint_{\Gamma_{\text{int}}} \{\{k\nabla_h T\}\} \cdot \mathbf{n}_{\mathfrak{I},\Gamma}\, [\![r]\!] + \{\{k\nabla_h r\}\} \cdot \mathbf{n}_{\mathfrak{I},\Gamma}\, [\![T]\!] \;\mathrm{d}S \\ &- \oint_{\Gamma_{\text{int}}} \eta\, [\![T]\!]\, [\![r]\!] \;\mathrm{d}S + s_{H,\mathfrak{I}}(T,r), \end{aligned} \tag{6.11}$$

where $s_{H,\mathfrak{I}}(-,-)$ enforces $T = T_{\text{sat}}$ on $\mathfrak{I}$ with

$$\begin{aligned} s_{H,\mathfrak{I}}(T,r) = &+ \oint_{\mathfrak{I}} [\![k\,(\nabla_h T \cdot \mathbf{n}_{\mathfrak{I}})\, r]\!] + [\![k\,(\nabla_h r \cdot \mathbf{n}_{\mathfrak{I}})\, T]\!] \\ &- [\![k\,(\nabla_h r \cdot \mathbf{n}_{\mathfrak{I}})]\!]\, T_{\text{sat}} - \eta\, [\![kTr]\!] + \eta\, [\![kT]\!]\, T_{\text{sat}} \,\mathrm{d}S, \end{aligned} \tag{6.12}$$

Note that in this case the energy jump condition (2.35) is not explicitly enforced at the interface. This condition is used to compute the evaporative mass flux $\dot{m}$.

Finally, we specify the terms on the right hand side of the variational formulation (6.7), which summarises the Dirichlet and Neumann boundary conditions. The first term $t_H(r)$ describes the Dirichlet boundary condition of the convective part

$$t_H(r) = -\oint_{\Gamma_{\text{D}}} \rho c \left((T_{\text{D}} \mathbf{u}_{\text{D}}) \cdot \mathbf{n}_{\Gamma} + \frac{\lambda}{2} T_{\text{D}} \right) r \,\mathrm{d}S. \tag{6.13}$$

The discretizations of the Dirichlet and Neumann boundary conditions for the conductive parts $s_H(r)$, are given as

$$s_H(r) = -\oint_{\Gamma_{\text{D}}} T_{\text{D}}\, (\nabla_h r \cdot \mathbf{n}_{\Gamma} - \eta r) \;\mathrm{d}S + \oint_{\Gamma_{\text{N}}} \mathbf{q}_{\text{N}} \cdot \mathbf{n}_{\Gamma} r \,\mathrm{d}S. \tag{6.14}$$

6.2. Coupling the Heat Equation with the Multi-Phase Flow Solver

For the coupling of the heat equation with the entire flow solver, we employ a strong coupling for the velocity parameter $\mathbf{u}^*$ of the heat equations. This means that the velocity $\mathbf{u}^*$ in the heat convection terms (6.9) and (6.10) is equal to the linearization velocity for the convective terms in the Navier-Stokes discretization (3.40). Both the Navier-Stokes equations and the heat equation are solved in a monolithic linear system for every non-linear iteration.

The discretization above assumes that the interface temperature is constant with $T = T_{\text{sat}}$. Thus, no enforcement of the energy jump condition (2.35) is given. This condition is used to evaluate the evaporative mass flux with

$$\dot{m} = \frac{1}{h_{\text{vap}}} \left(\mathbf{q}_{\mathfrak{B}} \cdot \mathbf{n}_{\mathfrak{I}} - \mathbf{q}_{\mathfrak{A}} \cdot \mathbf{n}_{\mathfrak{I}} \right) = \frac{1}{h_{\text{vap}}} \left(-k_{\mathfrak{B}} \nabla T_{\mathfrak{B}} \cdot \mathbf{n}_{\mathfrak{I}} + k_{\mathfrak{A}} \nabla T_{\mathfrak{A}} \cdot \mathbf{n}_{\mathfrak{I}} \right). \tag{6.15}$$

The computation of the mass flux is decoupled from the non-linear flow solver process, such as the interface evolution. We set $\dot{m} = \dot{m}|_{t^n}$ in the extended Navier-Stokes discretization (6.1). Thus, we ensure that the interface evolution and consequently the evolution velocity is always in accordance with the corresponding mass flux, since $\mathbf{w} = \mathbf{u}_\mathfrak{s} + \frac{\dot{m}}{\rho_\mathfrak{s}}\mathbf{n}_\mathfrak{I}$, see Equation (2.29). Again, we generate a density averaged velocity field for the advection equation with

$$\mathbf{w}_{\rho\text{Aver}} = \frac{\rho_\mathfrak{A}\mathbf{w}_\mathfrak{A} + \rho_\mathfrak{B}\mathbf{w}_\mathfrak{B}}{\rho_\mathfrak{A} + \rho_\mathfrak{B}}. \tag{6.16}$$

The extension of the velocity values at the interface is then constructed by the elliptic extension velocity, see Equations (3.92).

6.3. Numerical Results - Liquid-Vapour Phase-Change Problems

We are considering phase-change problems for liquid-vapour systems following the test cases proposed by Lemonnier et al. (2005). This work provides analytical solutions for one-dimensional problems, where the vapour phase is in contact with a heated wall, see Figure 6.1. It is assumed for all times that the interface is planar and the only spatial coordinate z is

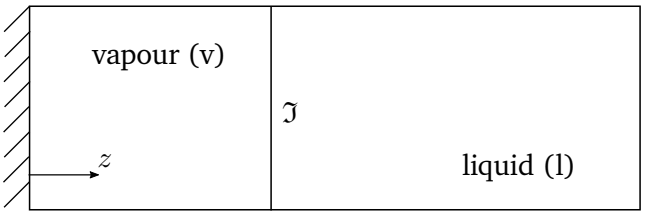


Figure 6.1.: Sketch of a one-dimensional liquid-vapour problem with the vapour-phase in contact with a wall

directed normal to the interface. Further, both phases are considered incompressible and the density does not depend on the temperature T. Thus, the Navier-Stokes equations (2.3) and the heat equation (2.9) reduce to

$$\begin{aligned} \rho\left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial z}\right) &= -\frac{\partial p}{\partial z}, \\ \frac{\partial u}{\partial z} &= 0, \\ \rho c\left(\frac{\partial T}{\partial t} + u\frac{\partial T}{\partial z}\right) &= \frac{\partial}{\partial z}\left(k\frac{\partial T}{\partial z}\right), \end{aligned} \tag{6.17}$$

where $u = u(t)$ denotes the velocity in interface normal direction. The corresponding interface jump conditions are given as described in Section 2.5, where the momentum jump reduces to

$$p_\mathrm{l} - p_\mathrm{v} = \dot{m}^2\left(\frac{1}{\rho_\mathrm{v}} - \frac{1}{\rho_\mathrm{l}}\right). \tag{6.18}$$

The subscripts l and v denote the liquid and vapour phase respectively. Closing the set of equations, the condition of local thermodynamic equilibrium of the interface, i.e. $T_\mathfrak{I} = T_\text{sat}(p_\mathfrak{I})$,

is added. Further details on the boundary conditions are given in the following sections. Two test cases for verification are considered: A quasi steady-state setup dominated by heat conduction and an unsteady setting, where the development of the thermal boundary layer near the interface is an key issue.

6.3.1. 1D quasi-steady state

Following Lemonnier et al. (2005), the first verification test case considers a system initially at rest and at uniform temperature $T(z, t = 0) = T_{\text{sat}}$. Imposing a heat flux q_{wall} at the wall z_{v} leads to vaporization at the interface, which consequently moves upward. This movement reaches a constant velocity for Pe $\ll 1$, where

$$\text{Pe} := \frac{\dot{m} c L}{k} \tag{6.19}$$

denotes the bulk Peclet number with the bulk length L of the corresponding phase. This dimensionless number describes the ratio between convective and conductive heat flux. Thus, for Pe $\ll 1$ the conductive terms dominate the given problem and information concerning the temperature T_{sat} at the interface have time to diffuse within the bulk phases without being perturbed by convection effects (Lemonnier et al., 2005). Thus, the conductive heat flux in both phases is constant and the energy jump condition (2.21) can be rewritten to

$$q_{\text{l}} - q_{\text{v}} = \dot{m} \left(h_{\text{vap}} + \frac{\dot{m}^2}{2} \left(\frac{1}{\rho_{\text{v}}^2} - \frac{1}{\rho_{\text{l}}^2} \right) \right), \tag{6.20}$$

where q_{l} and q_{v} denote the respective conductive heat flux in the bulk. Note that q_{v} is equal to the imposed heat flux q_{wall} at z_{v}. Thus, the temperature field within both bulk phases is linear with

$$T(z) \approx T_{\mathfrak{I}} + \frac{q}{k}(z_0 - z) \tag{6.21}$$

with z_0 representing the lower boundary for the vapour z_{v} phase and respectively the upper boundary for the liquid phase z_{l}. The system can be further simplified by ensuring that

$$h_{\text{vap}} \gg \frac{\dot{m}^2}{2} \left(\frac{1}{\rho_{\text{v}}^2} - \frac{1}{\rho_{\text{l}}^2} \right). \tag{6.22}$$

A result of this condition is that the interface temperature $T_{\mathfrak{I}}$ is approximately the saturation temperature at pressure p_0, i.e. $T_{\mathfrak{I}} \approx T_{\text{sat}}(p_0)$. Note that the pressure p_0 is imposed at the upper boundary z_{l}. Thus, one ends up with $q_{\text{l}} = 0$ and the interfacial mass flux results in

$$\dot{m} = -\frac{q_{\text{v}}}{h_{\text{vap}}}. \tag{6.23}$$

The corresponding constant interface velocity is given by

$$w = -\frac{\dot{m}}{\rho_{\text{v}}} \tag{6.24}$$

so the interface displacement $z_{\mathfrak{I}}$ reads

$$z_{\mathfrak{I}}(t) = z_{\mathfrak{I}}(0) + wt. \tag{6.25}$$

The velocity of the liquid phase is given by

$$u_\mathrm{l} = -\dot{m}\left(\frac{1}{\rho_\mathrm{v}} - \frac{1}{\rho_\mathrm{l}}\right). \tag{6.26}$$

For a detailed derivation the reader is referred to Lemonnier et al. (2005). According to Lemonnier et al. (2005), the model above can be used to verify how well a numerical method accounts for the interfacial mass and energy balance equations.

Setup description The above described test case is in the following implemented as a two-dimensional setup on the computational domain $\Omega = [0, L] \times [0, L]$ with $L = 0.1$. The initial interface position is set to $z_\mathfrak{I}(0) = 0.01$ in x-direction, see Figure 6.2. The left boundary is a no slip wall $\partial\Omega_\mathrm{wall}$ with an imposed heat flux of $q_\mathrm{wall} = 10$. The right boundary is an outlet with $p_0 = 10$ and a zero heat flux Neumann boundary condition. Both lower and upper boundary are periodic in y-direction.

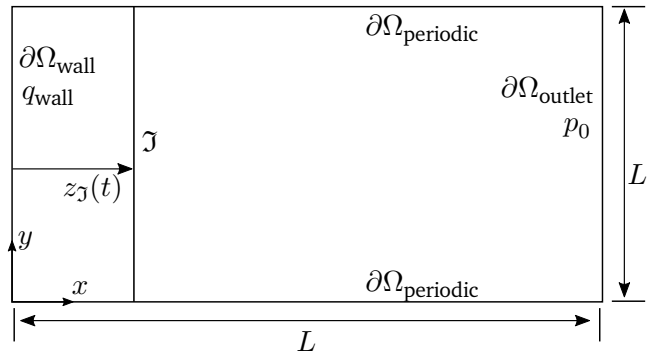


Figure 6.2.: Setup for the one-dimensional quasi-steady state phase-change test case

The physical parameters for the upper liquid phase are set to unity. Table 6.1 provides the parameters for the lower vapour phase. Note that the considered problem is independent of the viscosity, thus we set $\mu = 1$ for both phases. The corresponding Peclet-numbers are $\mathrm{Pe}_\mathrm{l} = 0.01$ for the liquid and $\mathrm{Pe}_\mathrm{v} = 1 \cdot 10^{-4}$ for the vapour phase. The domain size $L = 0.1$ was used to determine the upper bound for satisfying condition (6.22). Thus, the resulting mass flux (6.23) is given by $\dot{m} = -0.1$ and the corresponding interface normal velocity (6.24) by $w = 1.0$. The velocity of the liquid phase reads $u_\mathrm{l} = 0.9$. The initial temperature profile is set by Equation (6.21) with the saturation temperature of $T_\mathrm{sat} = 100$.

Table 6.1.: Physical parameters for the quasi steady state phase-change test case.

ρ_v	c_v	k_v	$T_\mathrm{sat}(p_0)$	h_vap	q_wall	L
0.1	0.001	0.1	100	100	10	0.1

Comparison to analytical solution For the comparison to the analytical solution, the domain is discretized using a 17×17 grid with additional adaptive mesh refinement (level 1) at the interface. The polynomial degree is set to 2, i.e. $k_\mathbf{u} = k_T = 2$ for the velocity fields and temperature field and $k_p = 1$ for the pressure field. The time step size is $\Delta t = 5 \cdot 10^{-4}$. In

Figure 6.3 the temperature profile (left) and velocity profile (right) at $y = 0.42$ is depicted for times $t = \{0, 2 \cdot 10^{-2}, 4 \cdot 10^{-2}, 6 \cdot 10^{-2}, 8 \cdot 10^{-2}\}$. One should note the kink in the temperature field and the jump in the velocity field at the position of the interface. The interface moves along the positive x-direction with $z_\mathfrak{I}(t) = 0.01 + t$. Both results show excellent agreement to the analytical solution over all times.

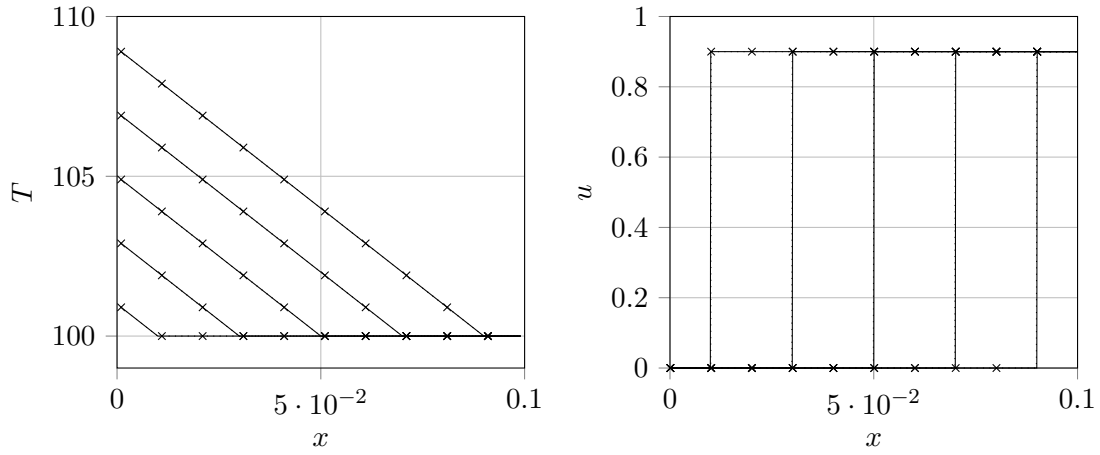


Figure 6.3.: Temperature (left) and velocity profiles (right) for the 1D quasi-steady state liquid-vapour phase-change problem at times $t = \{0, 2 \cdot 10^{-2}, 4 \cdot 10^{-2}, 6 \cdot 10^{-2}, 8 \cdot 10^{-2}\}$. The numerical solutions are described by the dotted lines with markers set every 100th data point. The solid lines denote the exact solutions for the temperature (6.21) and the velocity of the liquid phase $u_\mathrm{l} = 0.9$.

6.3.2. 1D unsteady

The second test case considers a system initially at rest, but the vapour temperature is different from the saturation temperature with $T_\mathrm{v} = T_\mathrm{sat}(p_0) + \Delta T$. Thus, the vapour is not in a thermodynamic equilibrium state at the interface and in order to return to equilibrium an infinite temperature gradient develops, which diffuses in time through the vapour phase. A derivation of the corresponding vapour temperature field is given in Lemonnier et al. (2005) and the reader is referred to it for further details. The temperature field in the frame of reference linked to the interface reads

$$T_\mathrm{v}(\zeta, t) = T_\mathrm{sat} + \Delta T \frac{\mathrm{erf}(\lambda_\mathrm{v}) - \mathrm{erf}(\lambda_\mathrm{v} + \sqrt{\frac{1}{\alpha_\mathrm{v}}} \frac{\zeta}{2\sqrt{t}})}{1 + \mathrm{erf}(\lambda_\mathrm{v})}, \tag{6.27}$$

where λ_v needs to fulfil the non-linear equation

$$\frac{c_\mathrm{v}}{h_\mathrm{vap}} \Delta T = \sqrt{\frac{\pi}{2}} \lambda_\mathrm{v} \mathrm{e}^{\lambda_\mathrm{v}^2} (1 + \mathrm{erf}(\lambda_\mathrm{v})). \tag{6.28}$$

The interface frame coordinate ζ is given by

$$\zeta = z - \int_0^t w(\tau) \, \mathrm{d}\tau \tag{6.29}$$

and the interface velocity for the unsteady test case reads

$$w(t) = \sqrt{\alpha_\mathrm{v}} \frac{\lambda_\mathrm{v}}{\sqrt{t}}. \tag{6.30}$$

Integration of (6.30) leads to the position of the interface with

$$z_\mathfrak{I}(t) = z_\mathfrak{I}(0) + 2\lambda_\mathrm{v}\sqrt{\alpha_\mathrm{v} t}. \tag{6.31}$$

The quantity $\alpha_\mathrm{v} = \frac{k_\mathrm{v}}{\rho_\mathrm{v} c_\mathrm{v}}$ is denoted as the thermal diffusivity.

Setup description For a numerical verification setup we follow the example of Lemonnier et al. (2005). We consider a similar two-dimensional setup with a planar interface in y-direction as described for the first test case, see Figure 6.2. The computational domain is set to $\Omega = [0, L_{v0} + L_{l0}] \times [0, L_{v0}/2]$, where L_{v0} denotes the initial length of the vapour phase, i.e. the initial interface position $z_\mathfrak{I}(0)$, and L_{l0} the length for the liquid phase, respectively. Those lengths are such chosen that the numerical results are comparable to the analytical solution with a precision of $\epsilon = 10^{-2}$ up to simulations times of $33\,\mathrm{s}$. This restriction is due to the validity of the analytical solution for infinite lengths (Lemonnier et al., 2005). On the left wall boundary a constant temperature of $T_\mathrm{v} = T_\mathrm{sat}(p_0) + \Delta T$ is imposed. The other boundaries are unchanged form the previous test case. The physical bulk parameters are given in Table 6.2.

Table 6.2.: Physical parameters for the unsteady phase-change test case.

phase	ρ in $\frac{\mathrm{kg}}{\mathrm{m}^3}$	c in $\frac{\mathrm{J}}{\mathrm{kg\,K}}$	k in $\frac{\mathrm{W}}{\mathrm{mK}}$	α in $\frac{\mathrm{m}^2}{\mathrm{s}}$
liquid	586.5	$9.35 \cdot 10^3$	0.444	$0.8096 \cdot 10^{-7}$
vapour	106.4	$15.4 \cdot 10^3$	0.114	$0.6958 \cdot 10^{-7}$

In this case water at a pressure of $p_0 = 160 \cdot 10^5$Pa is considered with a corresponding saturation temperature of $T_\mathrm{sat} = 620$K. The superheat at the wall is set to $\Delta T = 5$K and the heat of vaporization is $h_\mathrm{vap} = 941 \frac{\mathrm{J}}{\mathrm{kg}}$. Note that this value is not the experimental value but multiplied by a factor of 10^{-3}, so the interface moves with a comparable velocity as the development of the thermal boundary layer. The value for λ_v is given by 1.71814. Since the initial non-equilibrium state develops an infinite temperature gradient in the vapour phase, we impose the solution at $t_0 = 0.5 > 0$ as initial conditions for the temperature field (6.27). The pressure and velocity fields are set accordingly by Equation (6.18) and Equation (6.26).

Comparison to analytical solution The computational domain is discretized using a 17×4 grid with additional AMR level 3 at the interface. The polynomial degree is set to 3 ($k_\mathbf{u} = k_T = 3$ and $k_p = 2$) and the time step size is $\Delta t = 1 \cdot 10^{-2}$. Note that for this case the convective terms for the heat equation are not included, resulting in a decoupled computation of the flow and heat quantities. In Figure 6.4 the temperature profile at $y = 4.2 \cdot 10^{-3}$ is depicted for times $t = \{0, 1, 2, 3, 4, 5\}$. Note that the simulation is performed with AMR level 3 in order to allow the representation of the steep temperature gradient for $t = 0$. One can see the diffusion of this gradient in the vapour phase as the interface moves in positive x-direction. The results show very good agreement to the analytical solution.

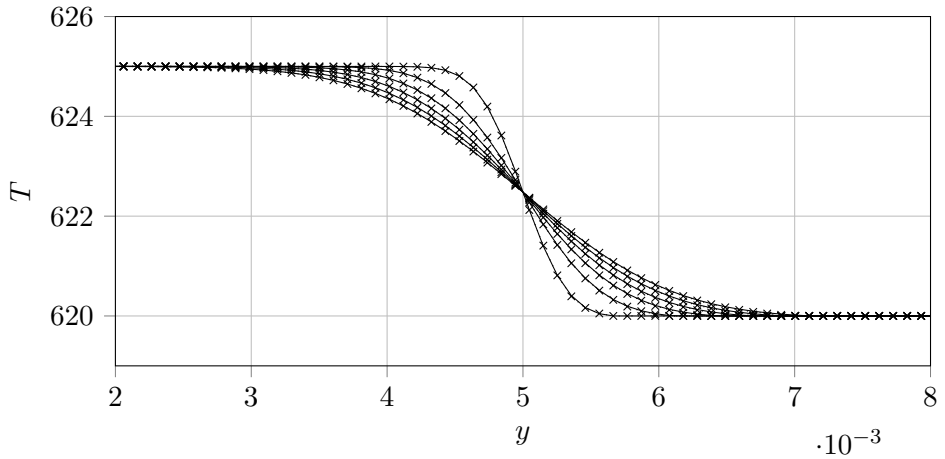


Figure 6.4.: Temperature profiles for the 1D unsteady liquid-vapour phase-change problem at times $t = \{0, 1, 2, 3, 4, 5\}$. The numerical solutions are described by the dotted lines with markers set every 10th data point. The solid lines denote the exact solutions for the temperature (6.27).

6.4. Conclusion

In this chapter the extension of the developed XNSE-solver to problems exhibiting evaporation at the interface was presented. Therefore, the XDG discretization of the Navier-Stokes equations were extended to allow a mass flux across the interface. Thus, the velocity field features a jump at the interface accounting for the mass flux. Furthermore, a two-phase heat equation discretization in the context of the XDG method was proposed, where the interface temperature is assumed to be equal to the imposed saturation temperature. The XDG discretization allows the sharp representation of the kink in the temperature field. Thus, the resulting jump in the heat flux field is used to evaluate the mass flux directly at the interface. The mass flux computation together with interface evolution are decoupled from the flow solver in order to construct the interface velocity accordingly. However, the solutions of the flow solver and the heat equation are strongly coupled.

A first verification of the presented extended XNSE-solver is provided by comparing to basic one-dimensional liquid-vapour phase-change problems. The numerical results showed excellent agreement. However, the considered test cases neglect the influence of the convective heat terms. Thus, more complex test cases need to be performed for a thorough verification and validation.

7. Conclusion and Outlook

In this work a high-order flow solver for the direct simulation of various multiphase flow problems was developed. This included the simulation of dynamic contact lines and evaporation problems. Thus, the three major goals of the presented work are stated as follows: first, the development of a validated multi-phase flow base solver in context of the XDG method, second, the extension to flow problems involving dynamic contact lines and third, the extension to problems exhibiting evaporation at the interface.

Regarding the base flow solver called XNSE-solver (eXtended-Navier-Stokes-Equations-solver), the governing equations were given in a sharp interface formulation and the discretization was done by using the XDG method. This allows the sub-cell accurate representation of sharp jumps in the pressure field and kinks in the velocity field resulting from the jump conditions at the interface. Furthermore, the spatial convergence rate of h^{k+1} is regained for such low-regularity solutions. Regarding the temporal discretization an XDG adapted moving interface approach was applied also allowing high-order convergence in time. The interface represented by a level-set function was described with a standard DG formulation accompanied by XDG specialized methods for reinitialization and the construction of an extension velocity for the level-set advection. Furthermore, an L^2-projection with continuity constraints was developed and implemented in order to ensure continuity of the interface for the adaption of the discretization. The numerical treatment of the surface tension force was implemented via the Laplace-Beltrami formulation without regularization. The XNSE-solver was validated against a wide range of typical two-phase surface tension driven flow phenomena including the analytical solution of capillary waves and benchmark results for a rising bubble.

For the extension to flow problems exhibiting dynamic contact lines at the domain boundaries, the generalized Navier boundary condition was adapted for the XDG discretization and successfully implemented within the XNSE-solver. The participation in establishing a new benchmark setup for the capillary rise which describes a prominent test case for capillarity driven flows is a highlight in this context. The benchmark originates from the Research Area B Modelling and Simulation within the CRC 1194 including four different numerical solver codes. The results are published in Gründing et al. (2020).

For the extension of the XNSE-solver for problems with additional heat transfer and evaporation, the two-phase heat equation was discretized by the XDG method. Furthermore, the XDG discretization for both the Navier-Stokes equations and the heat equation wass extended to allow a mass and energy flow across the interface. Note that in this case the velocity field exhibits a sharp jump and the temperature field shows a kink at the interface. First validation test cases against analytical solutions were presented.

At this point the author wants to give a remark on the implementational work done within the XNSE-solver and the BoSSS framework. One major part regards the extension of the XNSE-solver for adaptive mesh refinement and its various implemented options. Considering the soft-restart for each mesh adaption the reinitialization needs to be carefully performed in order to avoid numerical artifacts. Regarding the local refinement at the interface, the numerical quadrature had to be adapted to allow hanging nodes within the set of cut cells. Furthermore, the quadrature was adapted to allow the evaluation of the contact line in 3D. However, besides such necessary work a great achievement denotes the restructuring of the extended spatial operator called XSpatialOperator. This marks the core object of the XNSE-solver, where the spatial discretizations are implemented and evaluated. During the solver developments the XSpatialOperator was restructured such that it enables the handling of different discretization for each phase. Thus, it allows the combination of different physical regimes in each phase, e.g. a low-Mach solver and the incompressible flow solver. Furthermore, the above discretizations for the transient Navier-Stokes equations and heat equation are summarized in library-like classes. Discretizations for the kinetic energy balance equation are also provided by the author. In addition to that, a rudimentary XNSE-solver was extracted in order to be used as a base class for other works within the BoSSS-group. A successful application marks the two-phase adaption for the viscoelastic flow solver. However, the results showed that more work needs to be done regarding the stability at the interface.

An issue that needs further attention is the above mentioned high-order convergence in time by the moving interface approach. Although shown for the scalar advection, heat and Burgers' equations in Kummer et al. (2018), no rigorous temporal convergence study was provided in this work for the transient Navier-Stokes equations due to a lack of full analytical solutions for the two-phase case. A remedy for this may provide the novel unified transform method (Fokas, 2002; Fokas, 2008). Using this method the author derived an analytical solution for two-phase Couette flow with constant wall transpiration. The results are published in Smuda and Oberlack (2019). The unified method was also applied to non-linear droplet oscillations Plümacher et al. (2020).

All simulations presented in this work were two-dimensional and solved by direct solvers, but considering the simulation of three-dimensional problems the number of degrees of freedom requires the need of iterative solvers. First results for linear systems in context of XDG performed with a combined p- and h-multigrid method are published in Kummer et al. (2020). Newton-methods for the non-linear Navier-Stokes problem coupled with the level-set evolution are subject of ongoing work at the chair of fluid dynamics. Furthermore, an efficient parallelization of the interface evolution algorithm is needed for transient simulations in three dimensions.

Looking at the second approved funding period of the CRC 1194, the focus is on the investigation of contact line problems in combination with heat transfer. First successful tests were already performed, but a thorough validation needs to be conducted and additional steps have to be taken: implementation of the conjugate heat transfer between fluid and wall, introducing non-linear convective terms due to high evaporation rates and extension of the surface tension in dependence of the interface temperature.

Bibliography

Adalsteinsson, D. and Sethian, J. (1999). “The Fast Construction of Extension Velocities in Level Set Methods”. en. In: *Journal of Computational Physics* 148.1, pp. 2–22. ISSN: 00219991. DOI: 10.1006/jcph.1998.6090.

Adalsteinsson, D. and Sethian, J. A. (1995). “A Fast Level Set Method for Propagating Interfaces”. en. In: *Journal of Computational Physics* 118.2, pp. 269–277. ISSN: 00219991. DOI: 10.1006/jcph.1995.1098.

Amestoy, P. R., Duff, I. S., L’Excellent, J.-Y., and Koster, J. (2001). “A Fully Asynchronous Multifrontal Solver Using Distributed Dynamic Scheduling”. en. In: *SIAM J. Matrix Anal. & Appl.* 23.1, pp. 15–41. ISSN: 0895-4798, 1095-7162. DOI: 10.1137/S0895479899358194.

Amestoy, P. R., Guermouche, A., L’Excellent, J.-Y., and Pralet, S. (2006). “Hybrid scheduling for the parallel solution of linear systems”. en. In: *Parallel Computing* 32.2, pp. 136–156. ISSN: 01678191. DOI: 10.1016/j.parco.2005.07.004.

Arnold, D. N. (1982). “An Interior Penalty Finite Element Method with Discontinuous Elements”. en. In: *SIAM Journal on Numerical Analysis* 19.4, pp. 742–760. ISSN: 0036-1429, 1095-7170. DOI: 10.1137/0719052.

Aslam, T., Luo, S., and Zhao, H. (2014). “A static PDE Approach for MultiDimensional Extrapolation Using Fast Sweeping Methods”. en. In: *SIAM J. Sci. Comput.* 36.6, A2907–A2928. ISSN: 1064-8275, 1095-7197. DOI: 10.1137/140956919.

Babuška, I. and Banerjee, U. (2012). “Stable Generalized Finite Element Method (SGFEM)”. en. In: *Computer Methods in Applied Mechanics and Engineering* 201-204, pp. 91–111. ISSN: 00457825. DOI: 10.1016/j.cma.2011.09.012.

Babuška, I. (1973). “The finite element method with Lagrangian multipliers”. en. In: *Numerische Mathematik* 20.3, pp. 179–192. ISSN: 0029-599X, 0945-3245. DOI: 10.1007/BF01436561.

Bänsch, E. (2001). “Finite element discretization of the Navier–Stokes equations with a free capillary surface:” en. In: *Numer. Math.* 88.2, pp. 203–235. ISSN: 0029-599X. DOI: 10.1007/PL00005443.

Bao, K., Shi, Y., Sun, S., and Wang, X.-P. (2012). “A finite element method for the numerical solution of the coupled Cahn–Hilliard and Navier–Stokes system for moving contact line problems”. en. In: *Journal of Computational Physics* 231.24, pp. 8083–8099. ISSN: 00219991. DOI: 10.1016/j.jcp.2012.07.027.

Bastian, P. and Engwer, C. (2009). “An unfitted finite element method using discontinuous Galerkin”. en. In: *International Journal for Numerical Methods in Engineering* 79.12, pp. 1557–1576. ISSN: 00295981, 10970207. DOI: 10.1002/nme.2631.

Basting, C. and Kuzmin, D. (2013). “A minimization-based finite element formulation for interface-preserving level set reinitialization”. en. In: *Computing* 95.S1, pp. 13–25. ISSN: 0010-485X, 1436-5057. DOI: 10.1007/s00607-012-0259-z.

Beck, L. (2018). “Numerical Integration over implicitly defined Surfaces and Volumes”. MA thesis. TU Darmstadt.

Bernauer, M. K. and Herzog, R. (2012). “Implementation of an X-FEM Solver for the Classical Two-Phase Stefan Problem”. en. In: *J Sci Comput* 52.2, pp. 271–293. ISSN: 0885-7474, 1573-7691. DOI: 10.1007/s10915-011-9543-x.

Brackbill, J., Kothe, D., and Zemach, C. (1992). “A continuum method for modeling surface tension”. en. In: *Journal of Computational Physics* 100.2, pp. 335–354. ISSN: 00219991. DOI: 10.1016/0021-9991(92)90240-Y.

Brezzi, F. (1974). “On the existence, uniqueness and approximation of saddle-point problems arising from lagrangian multipliers”. en. In: *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique* 8.R2, pp. 129–151.

Brezzi, F., Cockburn, B., Marini, L., and Süli, E. (2006). “Stabilization mechanisms in discontinuous Galerkin finite element methods”. en. In: *Computer Methods in Applied Mechanics and Engineering* 195.25-28, pp. 3293–3310. ISSN: 00457825. DOI: 10.1016/j.cma.2005.06.015.

Burman, E. (2010). “Ghost penalty”. en. In: *Comptes Rendus Mathematique* 348.21-22, pp. 1217–1220. ISSN: 1631073X. DOI: 10.1016/j.crma.2010.10.006.

Buscaglia, G. C. and Ausas, R. F. (2011). “Variational formulations for surface tension, capillarity and wetting”. en. In: *Computer Methods in Applied Mechanics and Engineering* 200.45-46, pp. 3011–3025. ISSN: 00457825. DOI: 10.1016/j.cma.2011.06.002.

Chen, Q., Ramé, E., and Garoff, S. (1997a). “The velocity field near moving contact lines”. en. In: *J. Fluid Mech.* 337, pp. 49–66. ISSN: 0022-1120, 1469-7645. DOI: 10.1017/S0022112096004806.

Chen, S., Merriman, B., Osher, S., and Smereka, P. (1997b). “A Simple Level Set Method for Solving Stefan Problems”. en. In: *Journal of Computational Physics* 135.1, pp. 8–29. ISSN: 00219991. DOI: 10.1006/jcph.1997.5721.

Chessa, J., Smolinski, P., and Belytschko, T. (2002). “The extended finite element method (XFEM) for solidification problems”. en. In: *Int. J. Numer. Meth. Engng.* 53.8, pp. 1959–1977. ISSN: 0029-5981, 1097-0207. DOI: 10.1002/nme.386.

Chunming Li, Chenyang Xu, Changfeng Gui, and Fox, M. (2005). “Level Set Evolution without Re-Initialization: A New Variational Formulation”. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*. Vol. 1. San Diego, CA, USA: IEEE, pp. 430–436. ISBN: 978-0-7695-2372-9. DOI: 10.1109/CVPR.2005.213.

Cockburn, B., Kanschat, G., and Schötzau, D. (2004). “A locally conservative LDG method for the incompressible Navier-Stokes equations”. en. In: *Mathematics of Computation* 74.251, pp. 1067–1096. ISSN: 0025-5718. DOI: 10.1090/S0025-5718-04-01718-1.

Cockburn, B., Kanschat, G., and Schötzau, D. (2005). “The local discontinuous Galerkin method for linearized incompressible fluid flow: a review”. en. In: *Computers & Fluids* 34.4-5, pp. 491–506. ISSN: 00457930. DOI: 10.1016/j.compfluid.2003.08.005.

Cockburn, B. and Shu, C.-W. (1989). “TVB Runge-Kutta Local Projection Discontinuous Galerkin Finite Element Method for Conservation Laws II: General Framework”. In: *Mathematics of Computation* 52.186, p. 411. ISSN: 00255718. DOI: 10.2307/2008474.

Cockburn, B. and Shu, C.-W. (1998). “The Local Discontinuous Galerkin Method for Time-Dependent Convection-Diffusion Systems”. en. In: *SIAM Journal on Numerical Analysis* 35.6, pp. 2440–2463. ISSN: 0036-1429, 1095-7170. DOI: 10.1137/S0036142997316712.

Dahlquist, G. G. (1963). “A special stability problem for linear multistep methods”. en. In: *BIT* 3.1, pp. 27–43. ISSN: 0006-3835, 1572-9125. DOI: 10.1007/BF01963532.

Denner, F. and Wachem, B. G. van (2015). “Numerical time-step restrictions as a result of capillary waves”. en. In: *Journal of Computational Physics* 285, pp. 24–40. ISSN: 00219991. DOI: 10.1016/j.jcp.2015.01.021.

Deville, M. O., Fischer, P. F., and Mund, E. H. (2002). *High-order methods for incompressible fluid flow*. Cambridge monographs on applied and computational mathematics 9. Cambridge, UK ; New York: Cambridge University Press. ISBN: 978-0-521-45309-7.

Di Pietro, D. A. and Ern, A. (2012). *Mathematical Aspects of Discontinuous Galerkin Methods*. Vol. 69. Mathématiques et Applications. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-642-22979-4 978-3-642-22980-0.

Douglas, J. and Dupont, T. (1976). “Interior Penalty Procedures for Elliptic and Parabolic Galerkin Methods”. en. In: *Computing Methods in Applied Sciences*. Ed. by R. Glowinski and J. L. Lions. Vol. 58. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 207–216. ISBN: 978-3-540-08003-9 978-3-540-37550-0. DOI: 10.1007/BFb0120591.

Dupont, J.-B. and Legendre, D. (2010). “Numerical simulation of static and sliding drop with contact angle hysteresis”. en. In: *Journal of Computational Physics* 229.7, pp. 2453–2478. ISSN: 00219991. DOI: 10.1016/j.jcp.2009.07.034.

Dussan V., E. B. (1976). “The moving contact line: the slip boundary condition”. en. In: *J. Fluid Mech.* 77.4, pp. 665–684. ISSN: 0022-1120, 1469-7645. DOI: 10.1017/S0022112076002838.

Dussan V., E. B. and Davis, S. H. (1974). “On the motion of a fluid-fluid interface along a solid surface”. en. In: *J. Fluid Mech.* 65.1, pp. 71–95. ISSN: 0022-1120, 1469-7645. DOI: 10.1017/S0022112074001261.

Dziuk, G. (1990). “An algorithm for evolutionary surfaces.” In: *Numerische Mathematik* 58.6, pp. 603–612.

Fokas, A. S. (2002). “A new transform method for evolution partial differential equations”. en. In: *IMA Journal of Applied Mathematics* 67.6, pp. 559–590. ISSN: 0272-4960, 1464-3634. DOI: 10.1093/imamat/67.6.559.

Fokas, A. S. (2008). *A unified approach to boundary value problems*. eng. CBMS-NSF regional conference series in applied mathematics 78. OCLC: 612282841. Philadelphia, Pa: SIAM, Society for Industrial and Applied Mathematics. ISBN: 978-0-89871-651-1.

Fricke, M., Köhne, M., and Bothe, D. (2019). “A kinematic evolution equation for the dynamic contact angle and some consequences”. en. In: *Physica D: Nonlinear Phenomena* 394, pp. 26–43. ISSN: 01672789. DOI: 10.1016/j.physd.2019.01.008.

Fries, N. and Dreyer, M. (2008). “An analytic solution of capillary rise restrained by gravity”. en. In: *Journal of Colloid and Interface Science* 320.1, pp. 259–263. ISSN: 00219797. DOI: 10.1016/j.jcis.2008.01.009.

Fries, T. P. (2009). “The intrinsic XFEM for two-fluid flows”. en. In: *International Journal for Numerical Methods in Fluids* 60.4, pp. 437–471. ISSN: 02712091, 10970363. DOI: 10.1002/fld.1901.

Fries, T.-P. and Zilian, A. (2009). “On time integration in the XFEM”. en. In: *International Journal for Numerical Methods in Engineering* 79.1, pp. 69–93. ISSN: 00295981, 10970207. DOI: 10.1002/nme.2558.

Gallot, S., Hulin, D., and Lafontaine, J. (2004). *Riemannian Geometry*. Universitext. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-540-20493-0 978-3-642-18855-8. DOI: 10.1007/978-3-642-18855-8.

Ganesan, S. (2013). “On the dynamic contact angle in simulation of impinging droplets with sharp interface methods”. en. In: *Microfluid Nanofluid* 14.3-4, pp. 615–625. ISSN: 1613-4982, 1613-4990. DOI: 10.1007/s10404-012-1080-x.

Ganesan, S. and Tobiska, L. (2009). “Modelling and simulation of moving contact line problems with wetting effects”. en. In: *Comput. Visual Sci.* 12.7, pp. 329–336. ISSN: 1432-9360, 1433-0369. DOI: 10.1007/s00791-008-0111-3.

Gao, M. and Wang, X.-P. (2014). “An efficient scheme for a phase field model for the moving contact line problem with variable density and viscosity”. en. In: *Journal of Computational Physics* 272, pp. 704–718. ISSN: 00219991. DOI: 10.1016/j.jcp.2014.04.054.

Geisenhofer, M., Kummer, F., and Müller, B. (2019). “A discontinuous Galerkin immersed boundary solver for compressible flows: Adaptive local time stepping for artificial viscosity–based shock-capturing on cut cells”. en. In: *Int J Numer Meth Fluids* 91.9, pp. 448–472. ISSN: 0271-2091, 1097-0363. DOI: 10.1002/fld.4761.

Gerbeau, J.-F. and Lelièvre, T. (2009). “Generalized Navier boundary condition and geometric conservation law for surface tension”. en. In: *Computer Methods in Applied Mechanics and Engineering* 198.5-8, pp. 644–656. ISSN: 00457825. DOI: 10.1016/j.cma.2008.09.011.

Gibou, F. and Fedkiw, R. (2005). “A fourth order accurate discretization for the Laplace and heat equations on arbitrary domains, with applications to the Stefan problem”. en. In: *Journal of Computational Physics* 202.2, pp. 577–601. ISSN: 00219991. DOI: 10.1016/j.jcp.2004.07.018.

Girault, V., Rivière, B., and Wheeler, M. F. (2004). “A discontinuous Galerkin method with nonoverlapping domain decomposition for the Stokes and Navier-Stokes problems”. en. In: *Math. Comp.* 74.249, pp. 53–85. ISSN: 0025-5718. DOI: 10.1090/S0025-5718-04-01652-7.

Groß, S. and Reusken, A. (2007). “An extended pressure finite element space for two-phase incompressible flows with surface tension”. en. In: *Journal of Computational Physics* 224.1, pp. 40–58. ISSN: 00219991. DOI: 10.1016/j.jcp.2006.12.021.

Gross, S. and Reusken, A. (2007). “Finite Element Discretization Error Analysis of a Surface Tension Force in Two-Phase Incompressible Flows”. en. In: *SIAM J. Numer. Anal.* 45.4, pp. 1679–1700. ISSN: 0036-1429, 1095-7170. DOI: 10.1137/060667530.

Gross, S. and Reusken, A. (2011). *Numerical methods for two-phase incompressible flows*. Springer Series in Computational Mathematics 40. OCLC: ocn723107909. Berlin: Springer. ISBN: 978-3-642-19685-0.

Gründing, D., Smuda, M., Antritter, T., Fricke, M., Rettenmaier, D., Kummer, F., Stephan, P., Marschall, H., and Bothe, D. (2020). “A comparative study of transient capillary rise using direct numerical simulations”. en. In: *Applied Mathematical Modelling* 86, pp. 142–165. ISSN: 0307904X. DOI: 10.1016/j.apm.2020.04.020.

Gründing, D. (2020). “An Arbitrary Lagrangian-Eulerian Method for the Direct Numerical Simulation of Wetting Processes”. PhD thesis. TU Darmstadt.

Hansbo, A. and Hansbo, P. (2002). “An unfitted finite element method, based on Nitsche’s method, for elliptic interface problems”. en. In: *Computer Methods in Applied Mechanics and Engineering* 191.47-48, pp. 5537–5552. ISSN: 00457825. DOI: 10.1016/S0045-7825(02)00524-8.

Hansbo, P., Larson, M. G., and Zahedi, S. (2014). “A cut finite element method for a Stokes interface problem”. In: *arXiv:1205.5684 [math]*.

Heimann, F. (2013). “An Unfitted Higher-Order Discontinuous Galerkin Method for Incompressible Two-Phase Flow with Moving Contact Lines”. PhD thesis. Ruperto-Carola University of Heidelberg.

Heimann, F., Engwer, C., Ippisch, O., and Bastian, P. (2013). “An unfitted interior penalty discontinuous Galerkin method for incompressible Navier-Stokes two-phase flow: UDG FOR INCOMPRESSIBLE NAVIER-STOKES TWO-PHASE FLOW”. en. In: *International Journal for Numerical Methods in Fluids* 71.3, pp. 269–293. ISSN: 02712091. DOI: 10.1002/fld.3653.

Hesthaven, J. S. and Warburton, T. (2008). *Nodal Discontinuous Galerkin Methods*. Ed. by J. E. Marsden, L. Sirovich, and S. S. Antman. Vol. 54. Texts in Applied Mathematics. New York, NY: Springer New York. ISBN: 978-0-387-72065-4 978-0-387-72067-8.

Huh, C. and Scriven, L. (1971). “Hydrodynamic model of steady movement of a solid/liquid/fluid contact line”. en. In: *Journal of Colloid and Interface Science* 35.1, pp. 85–101. ISSN: 00219797. DOI: 10.1016/0021-9797(71)90188-3.

Hysing, S. (2006). “A new implicit surface tension implementation for interfacial flows”. en. In: *International Journal for Numerical Methods in Fluids* 51.6, pp. 659–672. ISSN: 02712091. DOI: 10.1002/fld.1147.

Hysing, S., Turek, S., Kuzmin, D., Parolini, N., Burman, E., Ganesan, S., and Tobiska, L. (2009). “Quantitative benchmark computations of two-dimensional bubble dynamics”. en. In: *International Journal for Numerical Methods in Fluids* 60.11, pp. 1259–1288. ISSN: 02712091, 10970363. DOI: 10.1002/fld.1934.

Jurin, J. (1719). “II. An account of some experiments shown before the Royal Society; with an enquiry into the cause of the ascent and suspension of water in capillary tubes.” la. In: *Philosophical Transactions of the Royal Society of London* 30.355, pp. 739–747. ISSN: 0261-0523, 2053-9223. DOI: 10.1098/rstl.1717.0026.

Kaviany, M. (2001). *Principles of convective heat transfer*. 2nd ed. Mechanical engineering series. New York: Springer. ISBN: 978-0-387-95162-1.

Kikker, A. and Kummer, F. (2018). “A High-Order Local Discontinuous Galerkin Scheme for Viscoelastic Fluid Flow”. en. In: *Recent Advances in Computational Engineering*. Ed. by M. Schäfer, M. Behr, M. Mehl, and B. Wohlmuth. Vol. 124. Cham: Springer International Publishing, pp. 51–61. ISBN: 978-3-319-93890-5 978-3-319-93891-2. DOI: 10.1007/978-3-319-93891-2_4.

Klein, B., Kummer, F., Keil, M., and Oberlack, M. (2015). “An extension of the SIMPLE based discontinuous Galerkin solver to unsteady incompressible flows: SIMPLE BASED DG SOLVER FOR UNSTEADY INCOMPRESSIBLE FLOWS”. en. In: *International Journal for Numerical Methods in Fluids* 77.10, pp. 571–589. ISSN: 02712091. DOI: 10.1002/fld.3994.

Klein, B., Müller, B., Kummer, F., and Oberlack, M. (2016). “A high-order discontinuous Galerkin solver for low Mach number flows: DISCONTINUOUS GALERKIN SOLVER FOR LOW MACH NUMBER FLOWS”. en. In: *International Journal for Numerical Methods in Fluids* 81.8, pp. 489–520. ISSN: 02712091. DOI: 10.1002/fld.4193.

Klein, B. (2015). “A high-order Discontinuous Galerkin solver for incompressible and low-Mach number flows”. PhD Thesis. Darmstadt: Technische Universität Darmstadt.

Klein, B., Kummer, F., and Oberlack, M. (2013). “A SIMPLE based discontinuous Galerkin solver for steady incompressible flows”. en. In: *Journal of Computational Physics* 237, pp. 235–250. ISSN: 00219991. DOI: 10.1016/j.jcp.2012.11.051.

Krämer-Eis, S. (2017). “A high-order discontinuous Galerkin method for unsteady compressible flows with immersed boundaries”. PhD Thesis. Darmstadt: Technische Universität Darmstadt.

Krause, D. (2019). “A Cut Cell Discontinuous Galerkin Method for Particulate Flows”. PhD Thesis. Darmstadt: Technische Universität Darmstadt.

Krause, D. and Kummer, F. (2017). “An incompressible immersed boundary solver for moving body flows using a cut cell discontinuous Galerkin method”. en. In: *Computers & Fluids* 153, pp. 118–129. ISSN: 00457930. DOI: 10.1016/j.compfluid.2017.05.008.

Kummer, F. (2012). “The BoSSS Discontinuous Galerkin Solver for Incompressible Fluid Dynamics and an Extension to Singular Equations”. englisch. Dissertation. Darmstadt: Technische Universität Darmstadt.

Kummer, F. (2016). “Extended discontinuous Galerkin methods for two-phase flows: the spatial discretization: XDG METHODS FOR TWO-PHASE FLOWS: SPATIAL DISCRETIZATION”. en. In: *International Journal for Numerical Methods in Engineering*. ISSN: 00295981. DOI: 10.1002/nme.5288.

Kummer, F., Müller, B., and Utz, T. (2018). “Time integration for extended discontinuous Galerkin methods with moving domains: Time integration for XDG methods with moving domains”. en. In: *International Journal for Numerical Methods in Engineering* 113.5, pp. 767–788. ISSN: 00295981. DOI: 10.1002/nme.5634.

Kummer, F. and Warburton, T. (2016). “Patch-Recovery Filters for Curvature in Discontinuous Galerkin-Based Level-Set Methods”. en. In: *Communications in Computational Physics* 19.02, pp. 329–353. ISSN: 1815-2406, 1991-7120. DOI: 10.4208/cicp.191114.140715a.

Kummer, F., Weber, J., and Smuda, M. (2020). “BoSSS: A package for multigrid extended discontinuous Galerkin methods”. en. In: *Computers & Mathematics with Applications*, S0898122120301917. ISSN: 08981221. DOI: 10.1016/j.camwa.2020.05.001.

Lehrenfeld, C. (2015). “The Nitsche XFEM-DG Space-Time Method and its Implementation in Three Space Dimensions”. en. In: *SIAM Journal on Scientific Computing* 37.1, A245–A270. ISSN: 1064-8275, 1095-7197. DOI: 10.1137/130943534.

Lehrenfeld, C. and Reusken, A. (2012). “Nitsche-XFEM with Streamline Diffusion Stabilization for a Two-Phase Mass Transport Problem”. en. In: *SIAM Journal on Scientific Computing* 34.5, A2740–A2759. ISSN: 1064-8275, 1095-7197. DOI: 10.1137/110855235.

Lehrenfeld, C. and Reusken, A. (2013). “Analysis of a Nitsche XFEM-DG Discretization for a Class of Two-Phase Mass Transport Problems”. en. In: *SIAM Journal on Numerical Analysis* 51.2, pp. 958–983. ISSN: 0036-1429, 1095-7170. DOI: 10.1137/120875260.

Lemonnier, H., Jamet, D., and Lebaigue, O. (2005). *Validation of advanced computational methods for multiphase flow*. English. OCLC: 780270353. ISBN: 978-1-56700-324-6.

Lesaint, P. and Raviart, P. A. (1974). “On a Finite Element Method for Solving the Neutron Transport Equation”. eng. In: *Publications mathématiques et informatique de Rennes* S4, pp. 1–40.

Lewy, H., Friedrichs, K., and Courant, R. (1928). “Über die partiellen Differenzengleichungen der mathematischen Physik”. In: *Mathematische Annalen* 100, pp. 32–74.

Li, B. Q. (2006). *Discontinuous finite elements in fluid dynamics and heat transfer*. Computational fluid and solid mechanics. London: Springer. ISBN: 978-1-85233-988-3 978-1-84628-205-8.

Liu, J.-G. and Shu, C.-W. (2000). “A High-Order Discontinuous Galerkin Method for 2D Incompressible Flows”. en. In: *Journal of Computational Physics* 160.2, pp. 577–596. ISSN: 00219991. DOI: 10.1006/jcph.2000.6475.

Lucas, R. (1918). “Ueber das Zeitgesetz des kapillaren Aufstiegs von Flüssigkeiten”. de. In: *Kolloid-Zeitschrift* 23.1, pp. 15–22. ISSN: 0303-402X, 1435-1536. DOI: 10.1007/BF01461107.

Lukyanov, A. V. and Pryer, T. (2017). “Hydrodynamics of Moving Contact Lines: Macroscopic versus Microscopic”. en. In: *Langmuir* 33.34, pp. 8582–8590. ISSN: 0743-7463, 1520-5827. DOI: 10.1021/acs.langmuir.7b02409.

Manservisi, S. and Scardovelli, R. (2009). “A variational approach to the contact angle dynamics of spreading droplets”. en. In: *Computers & Fluids* 38.2, pp. 406–424. ISSN: 00457930. DOI: 10.1016/j.compfluid.2008.05.001.

Marchandise, E., Geuzaine, P., Chevaugeon, N., and Remacle, J.-F. (2007). “A stabilized finite element method using a discontinuous level set approach for the computation of bubble dynamics”. en. In: *Journal of Computational Physics* 225.1, pp. 949–974. ISSN: 00219991. DOI: 10.1016/j.jcp.2007.01.005.

Marks, R. J. (1991). *Introduction to Shannon Sampling and Interpolation Theory*. en. Ed. by J. B. Thomas. Springer Texts in Electrical Engineering. New York, NY: Springer New York. ISBN: 978-1-4613-9710-6 978-1-4613-9708-3. DOI: 10.1007/978-1-4613-9708-3.

Martin, D., Chaouki, H., Robert, J.-L., Fafard, M., and Ziegler, D. (2016). “A XFEM Lagrange Multiplier Technique for Stefan Problems”. In: *Frontiers in Heat and Mass Transfer* 7. ISSN: 2151-8629. DOI: 10.5098/hmt.7.31.

Martin, D., Chaouki, H., Robert, J.-L., Ziegler, D., and Fafard, M. (2018). “A XFEM PHASE CHANGE MODEL WITH CONVECTION”. In: *Frontiers in Heat and Mass Transfer* 10. ISSN: 2151-8629. DOI: 10.5098/hmt.10.18.

Meister, A. (2015). *Numerik linearer Gleichungssysteme*. de. Wiesbaden: Springer Fachmedien Wiesbaden. ISBN: 978-3-658-07199-8 978-3-658-07200-1. DOI: 10.1007/978-3-658-07200-1.

Merle, R. and Dolbow, J. (2002). “Solving thermal and phase change problems with the eXtended finite element method”. In: *Computational Mechanics* 28.5, pp. 339–350. ISSN: 0178-7675, 1432-0924. DOI: 10.1007/s00466-002-0298-y.

Moës, N., Dolbow, J., and Belytschko, T. (1999). “A finite element method for crack growth without remeshing”. In: *International Journal for Numerical Methods in Engineering* 46.1, pp. 131–150. ISSN: 1097-0207. DOI: 10.1002/(SICI)1097-0207(19990910)46:1$<$131::AID-NME726$>$3.0.CO;2-J.

Müller, B., Krämer-Eis, S., Kummer, F., and Oberlack, M. (2017). “A high-order discontinuous Galerkin method for compressible flows with immersed boundaries: A high-order discontinuous Galerkin method for compressible flows with immersed boundaries”. en. In: *Int. J. Numer. Meth. Engng* 110.1, pp. 3–30. ISSN: 00295981. DOI: 10.1002/nme.5343.

Müller, B., Kummer, F., and Oberlack, M. (2013). “Highly accurate surface and volume integration on implicit domains by means of moment-fitting: ACCURATE INTEGRATION ON IMPLICIT DOMAINS BY MEANS OF MOMENT-FITTING”. en. In: *International Journal for Numerical Methods in Engineering* 96.8, pp. 512–528. ISSN: 00295981. DOI: 10.1002/nme.4569.

Müller, B. (2014). “Methods for higher order numerical simulations of complex inviscid fluids with immersed boundaries”. Dissertation. Darmstadt: Technische Universität Darmstadt.

Navier, C. L. M. H. (1823). “Mémoire sur les lois du mouvement des fluides”. In: *Mémoires de l’Académie Royale des Sciences de l’Institut de France* 6, pp. 389–440.

Nitsche, J. (1971). “Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind”. de. In: *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg* 36.1, pp. 9–15. ISSN: 0025-5858, 1865-8784. DOI: 10.1007/BF02995904.

Panton, R. L. (2013). *Incompressible Flow: Panton/Incompressible Flow 4E*. en. Hoboken, NJ, USA: John Wiley & Sons, Inc. ISBN: 978-1-118-71307-5 978-1-118-01343-4. DOI: 10.1002/9781118713075.

Plümacher, D., Oberlack, M., Wang, Y., and Smuda, M. (2020). "On a non-linear droplet oscillation theory via the unified method". en. In: *Physics of Fluids* 32.6, p. 067104. ISSN: 1070-6631, 1089-7666. DOI: 10.1063/5.0007341.

Popinet, S. (2009). "An accurate adaptive solver for surface-tension-driven interfacial flows". en. In: *Journal of Computational Physics* 228.16, pp. 5838–5866. ISSN: 00219991. DOI: 10.1016/j.jcp.2009.04.042.

Popinet, S. and Zaleski, S. (1999). "A front-tracking algorithm for accurate representation of surface tension". In: *International Journal for Numerical Methods in Fluids* 30.6, pp. 775–793.

Prosperetti, A. (1981). "Motion of two superposed viscous fluids". en. In: *Physics of Fluids* 24.7, p. 1217. ISSN: 00319171. DOI: 10.1063/1.863522.

Reed, W. H. and Hill, T. R. (1973). "Triangular mesh methods for the neutron transport equation". In:

Ren, W. and E, W. (2007). "Boundary conditions for the moving contact line problem". en. In: *Physics of Fluids* 19.2, p. 022101. ISSN: 1070-6631, 1089-7666. DOI: 10.1063/1.2646754.

Reusken, A. (2008). "Analysis of an extended pressure finite element space for two-phase incompressible flows". en. In: *Computing and Visualization in Science* 11.4-6, pp. 293–305. ISSN: 1432-9360, 1433-0369. DOI: 10.1007/s00791-008-0099-8.

Reusken, A., Xu, X., and Zhang, L. (2017). "Finite element methods for a class of continuum models for immiscible flows with moving contact lines: FINITE ELEMENT METHODS FOR IMMISCIBLE FLOWS WITH MOVING CONTACT LINES". en. In: *International Journal for Numerical Methods in Fluids* 84.5, pp. 268–291. ISSN: 02712091. DOI: 10.1002/fld.4349.

Rideal, E. K. (1922). "CVIII. *On* the flow of liquids under capillary pressure". en. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 44.264, pp. 1152–1159. ISSN: 1941-5982, 1941-5990. DOI: 10.1080/14786441008634082.

Sauerland, H. (2013). *An XFEM based sharp interface approach for two-phase and free-surface flows*. eng. Strömungsmechanik. OCLC: 869868386. München: Verl. Dr. Hut. ISBN: 978-3-8439-1232-7.

Sauerland, H. and Fries, T.-P. (2013). "The stable XFEM for two-phase flows". en. In: *Computers & Fluids* 87, pp. 41–49. ISSN: 00457930. DOI: 10.1016/j.compfluid.2012.10.017.

Saye, R. I. (2015). "High-Order Quadrature Methods for Implicitly Defined Surfaces and Volumes in Hyperrectangles". en. In: *SIAM J. Sci. Comput.* 37.2, A993–A1019. ISSN: 1064-8275, 1095-7197. DOI: 10.1137/140966290.

Saye, R. (2016). "Interfacial gauge methods for incompressible fluid dynamics". en. In: *Science Advances* 2.6, e1501869. ISSN: 2375-2548. DOI: 10.1126/sciadv.1501869.

Saye, R. (2017). "Implicit mesh discontinuous Galerkin methods and interfacial gauge methods for high-order accurate interface dynamics, with applications to surface tension dynamics, rigid body fluid–structure interaction, and free surface flow: Part II". en. In: *Journal of Computational Physics* 344, pp. 683–723. ISSN: 00219991. DOI: 10.1016/j.jcp.2017.05.003.

Schenk, O., Gärtner, K., and Fichtner, W. (1999). "Scalable parallel sparse factorization with left-right looking strategy on shared memory multiprocessors". In: *High-Performance Computing and Networking*. Ed. by G. Goos, J. Hartmanis, J. van Leeuwen, P. Sloot, M. Bubak, A.

Hoekstra, and B. Hertzberger. Vol. 1593. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 221–230. ISBN: 978-3-540-65821-4 978-3-540-48933-7. DOI: 10.1007/BFb0100583.

Schott, B., Rasthofer, U., Gravemeier, V., and Wall, W. A. (2015). "A face-oriented stabilized Nitsche-type extended variational multiscale method for incompressible two-phase flow: NITSCHE-TYPE EXTENDED VARIATIONAL MULTISCALE METHOD FOR TWO-PHASE FLOW". en. In: *International Journal for Numerical Methods in Engineering* 104.7, pp. 721–748. ISSN: 00295981. DOI: 10.1002/nme.4789.

Schott, B. (2017). "Stabilized Cut Finite Element Methods for Complex Interface Coupled Flow Problems". Dissertation. München: Technische Universität München.

Shahbazi, K. (2005). "An explicit expression for the penalty parameter of the interior penalty method". en. In: *Journal of Computational Physics* 205.2, pp. 401–407. ISSN: 00219991. DOI: 10.1016/j.jcp.2004.11.017.

Shahbazi, K., Fischer, P. F., and Ethier, C. R. (2007). "A high-order discontinuous Galerkin method for the unsteady incompressible Navier–Stokes equations". en. In: *Journal of Computational Physics* 222.1, pp. 391–407. ISSN: 00219991. DOI: 10.1016/j.jcp.2006.07.029.

Shen, J., Yang, X., and Yu, H. (2015). "Efficient energy stable numerical schemes for a phase field moving contact line model". en. In: *Journal of Computational Physics* 284, pp. 617–630. ISSN: 00219991. DOI: 10.1016/j.jcp.2014.12.046.

Shikhmurzaev, Y. (1993). "The moving contact line on a smooth solid surface". en. In: *International Journal of Multiphase Flow* 19.4, pp. 589–610. ISSN: 03019322. DOI: 10.1016/0301-9322(93)90090-H.

Shikhmurzaev, Y. (2006). "Singularities at the moving contact line. Mathematical, physical and computational aspects". In: *Physica D: Nonlinear Phenomena* 217, pp. 121–133. DOI: 10.1016/j.physd.2006.03.003.

Shikhmurzaev, Y. (2007). *Capillary flows with forming interfaces*.

Slattery, J. C., Sagis, L., and Oh, E.-S. (2007). *Interfacial transport phenomena*. 2nd ed. New York: Springer. ISBN: 978-0-387-38438-2 978-0-387-38442-9.

Smolianski, A. (2001). "Numerical Modeling of Two-Fluid Interfacial Flows". PhD thesis. University of Jyväskylä.

Smuda, M. and Oberlack, M. (2019). "On the analytical solution of the two-phase Couette flow with wall transpiration". en. In: *Physics of Fluids* 31.12, p. 123603. ISSN: 1070-6631, 1089-7666. DOI: 10.1063/1.5119795.

Sprittles, J. E. and Shikhmurzaev, Y. D. (2011). "Viscous flows in corner regions: Singularities and hidden eigensolutions". en. In: *International Journal for Numerical Methods in Fluids* 65.4, pp. 372–382. ISSN: 02712091. DOI: 10.1002/fld.2187.

Stepanov, S., Vasilyeva, M., and Vasil'ev, V. I. (2018). "Generalized multiscale discontinuous Galerkin method for solving the heat problem with phase change". en. In: *Journal of Computational and Applied Mathematics* 340, pp. 645–652. ISSN: 03770427. DOI: 10.1016/j.cam.2017.12.004.

Sussman, M., Smereka, P., and Osher, S. (1994). "A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow". en. In: *Journal of Computational Physics* 114.1, pp. 146–159. ISSN: 00219991. DOI: 10.1006/jcph.1994.1155.

Utz, T. (2018). "Level set methods for high-order unfitted discontinuous Galerkin schemes". PhD thesis. TU Darmstadt.

Utz, T. and Kummer, F. (2018). “A high-order discontinuous Galerkin method for extension problems: A high-order DG method for extension problems”. en. In: *Int J Numer Meth Fluids* 86.8, pp. 509–518. ISSN: 02712091. DOI: 10.1002/fld.4464.

Utz, T., Kummer, F., and Oberlack, M. (2017). “Interface-preserving level-set reinitialization for DG-FEM: INTERFACE-PRESERVING REINITIALIZATION FOR DG”. en. In: *Int. J. Numer. Meth. Fluids* 84.4, pp. 183–198. ISSN: 02712091. DOI: 10.1002/fld.4344.

Wang, Y. and Oberlack, M. (2011). “A thermodynamic model of multiphase flows with moving interfaces and contact line”. en. In: *Continuum Mech. Thermodyn.* 23.5, pp. 409–433. ISSN: 0935-1175, 1432-0959. DOI: 10.1007/s00161-011-0186-9.

Washburn, E. W. (1921). “The Dynamics of Capillary Flow”. en. In: *Physical Review* 17.3, pp. 273–283. ISSN: 0031-899X. DOI: 10.1103/PhysRev.17.273.

Young, T. (1805). “III. An essay on the cohesion of fluids”. en. In: *Phil. Trans. R. Soc.* 95, pp. 65–87. ISSN: 0261-0523, 2053-9223. DOI: 10.1098/rstl.1805.0005.

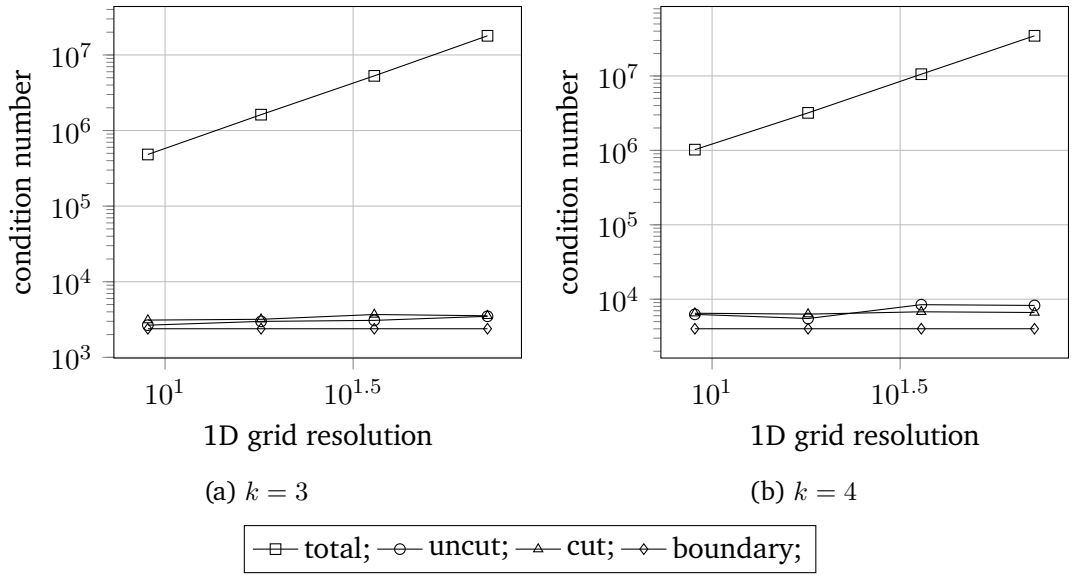
Zabaras, N., Ganapathysubramanian, B., and Tan, L. (2006). “Modelling dendritic solidification with melt convection using the extended finite element method”. en. In: *Journal of Computational Physics* 218.1, pp. 200–227. ISSN: 00219991. DOI: 10.1016/j.jcp.2006.02.002.

Zhao, H. (2004). “A fast sweeping method for Eikonal equations”. en. In: *Math. Comp.* 74.250, pp. 603–628. ISSN: 0025-5718. DOI: 10.1090/S0025-5718-04-01678-3.

Zienkiewicz, O. and Zhu, J. (1995). “Superconvergence and the superconvergent patch recovery”. en. In: *Finite Elements in Analysis and Design* 19.1-2, pp. 11–23. ISSN: 0168874X. DOI: 10.1016/0168-874X(94)00054-J.

A. Additional Numerical Results

A.1. Droplet in Steady-State

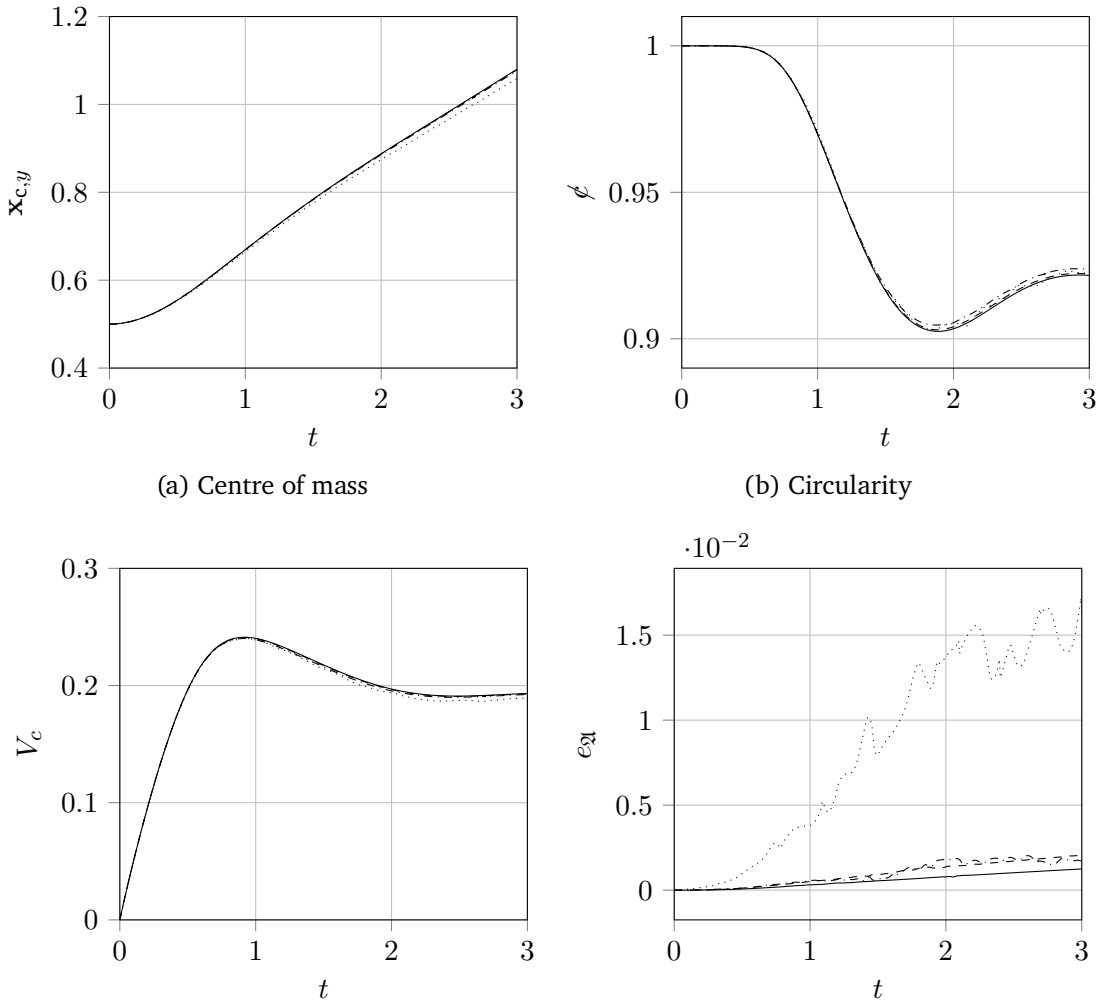


(a) $k = 3$ (b) $k = 4$

Figure A.1.: Condition number study for $k = \{3, 4\}$ of a steady-state droplet on the following meshes: $9\times9, 18\times18, 36\times36, 72\times72, 144\times144$. Considered are the total condition number of the entire system (square) and the maximum stencil condition number of inner uncut (circle) and cut cells (triangle) and boundary cells (diamond).

A.2. Rising Bubble Benchmark

Table A.1.: l_1-error norms and ROC_1 for the mesh study of the rising bubble benchmark.



(a) Centre of mass

(b) Circularity

(c) Rise velocity

(d) relative error mass

Figure A.2.: Convergence study for $k = 3$ of the temporal evolution of the benchmark quantities for test case 1: (a) y-component of the centre of mass $\mathbf{x}_c$ (4.14), (b) circularity ϕ (4.15), (c) rise velocity V_c and (d) relative error of the bubble area $e_{\mathfrak{A}} = \left|\frac{\int_{\mathfrak{A}} 1 \, \mathrm{d}V - 2\pi r_0^2}{2\pi r_0^2}\right|$.

	Center of mass		Rise velocity		Circularity	
$\frac{1}{h}$	$\|e\|_1$	ROC$_1$	$\|e\|_1$	ROC$_1$	$\|e\|_1$	ROC$_1$
$k = 2$						
20	$4.05 \cdot 10^{-3}$	-	$9.11 \cdot 10^{-3}$	-	$4.95 \cdot 10^{-4}$	-
40	$1.04 \cdot 10^{-3}$	1.96	$2.69 \cdot 10^{-3}$	1.76	$4.07 \cdot 10^{-4}$	0.28
60	$3.73 \cdot 10^{-4}$	2.53	$9.51 \cdot 10^{-4}$	2.56	$1.40 \cdot 10^{-4}$	2.63
$k = 3$						
10	$1.21 \cdot 10^{-2}$	-	$1.37 \cdot 10^{-2}$	-	$7.29 \cdot 10^{-4}$	-
20	$1.78 \cdot 10^{-3}$	2.76	$5.41 \cdot 10^{-3}$	1.34	$1.25 \cdot 10^{-3}$	-0.78
40	$5.59 \cdot 10^{-4}$	1.67	$1.53 \cdot 10^{-3}$	1.82	$3.45 \cdot 10^{-4}$	1.86

Table A.2.: l_2-error norms and ROC_2 for the mesh study of the rising bubble benchmark.

	Center of mass		Rise velocity		Circularity	
$\frac{1}{h}$	$\|e\|_2$	ROC_2	$\|e\|_2$	ROC_2	$\|e\|_2$	ROC_2
$k=2$						
20	$4.92 \cdot 10^{-3}$	-	$1.02 \cdot 10^{-2}$	-	$8.77 \cdot 10^{-4}$	-
40	$1.27 \cdot 10^{-3}$	1.95	$2.95 \cdot 10^{-3}$	1.79	$5.49 \cdot 10^{-4}$	0.68
60	$4.53 \cdot 10^{-4}$	2.54	$1.05 \cdot 10^{-3}$	2.55	$1.92 \cdot 10^{-4}$	2.59
$k=3$						
10	$1.44 \cdot 10^{-2}$	-	$1.56 \cdot 10^{-2}$	-	$9.58 \cdot 10^{-4}$	-
20	$2.23 \cdot 10^{-3}$	2.69	$5.96 \cdot 10^{-3}$	1.39	$1.65 \cdot 10^{-3}$	-0.78
40	$6.85 \cdot 10^{-4}$	1.70	$1.69 \cdot 10^{-3}$	1.82	$4.67 \cdot 10^{-4}$	1.04

Table A.3.: Benchmark quantities at distinct values in time for $k=2$ (test case 1)

$\frac{1}{h}$	20	40	60	80	benchmark groups
ϕ_{min}	0.9016	0.9027	0.9032	0.9021	0.9011 - 0.9013
$t\|_{\phi=\phi_{\text{min}}}$	1.870	1.881	1.887	1.890	1.875 - 1.9041
$V_{c,\text{max}}$	0.2402	0.2409	0.2412	0.2413	0.2417 - 0.2421
$t\|_{V_c=V_{c,\text{max}}}$	0.910	0.915	0.918	0.918	0.9213 - 0.9313
$y_c(t=3)$	1.0732	1.0785	1.0797	1.080	1.0799 - 1.0817

Table A.4.: Benchmark quantities at distinct values in time for $k=3$ (test case 1)

$\frac{1}{h}$	10	20	40	60	benchmark groups
ϕ_{min}	0.903	0.9047	0.9031	0.9025	0.9011 - 0.9013
$t\|_{\phi=\phi_{\text{min}}}$	1.854	1.869	1.878	1.887	1.875 - 1.9041
$V_{c,\text{max}}$	0.2404	0.2401	0.2409	0.2412	0.2417 - 0.2421
$t\|_{V_c=V_{c,\text{max}}}$	0.900	0.912	0.915	0.918	0.9213 - 0.9313
$y_c(t=3)$	1.0578	1.0768	1.0790	1.0800	1.0799 - 1.0817

Table A.5.: Benchmark quantities at distinct values in time for $k=2$ (test case 2)

$\frac{1}{h}$	20 + AMR 1	40 + AMR 1	80 + AMR 1	benchmark groups
ϕ_{min}	0.5507	0.5160	0.5093	0.4647 - 0.5869
$t\|_{\phi=\phi_{\text{min}}}$	2.9610	2.9220	2.9850	2.4004 - 3.000
$V_{c,\text{max1}}$	0.2492	0.2497	0.2500	0.2502 - 0.2524
$t\|_{V_c=V_{c,\text{max1}}}$	0.720	0.726	0.729	0.7281 - 0.7332
$V_{c,\text{max2}}$	0.2363	0.2352	0.2373	0.2393 - 0.2440
$t\|_{V_c=V_{c,\text{max2}}}$	2.070	2.019	2.020	1.9844 - 2.0705
$y_c(t=3)$	1.0964	1.1227	1.1300	1.1249 - 1.1380

A.3. Droplet on Slip Wall

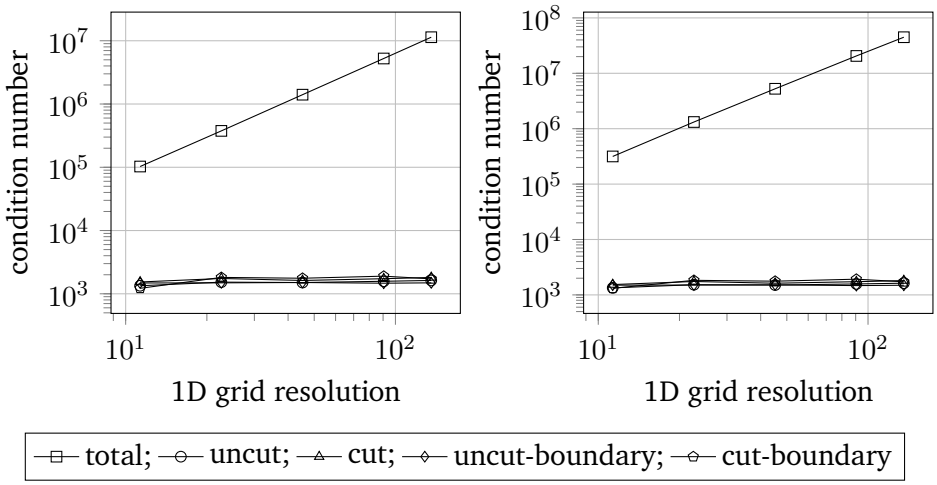


Figure A.3.: Condition number study for $\theta_{\text{stat}} = 90$ of a steady-state droplet on a slip wall with $\beta_S = 5$ (left) and $\beta_S = 0$, i.e. free-slip, (right). The study is performed on the following meshes: 8×16, 16×32, 32×64, 64×128, 96×192.

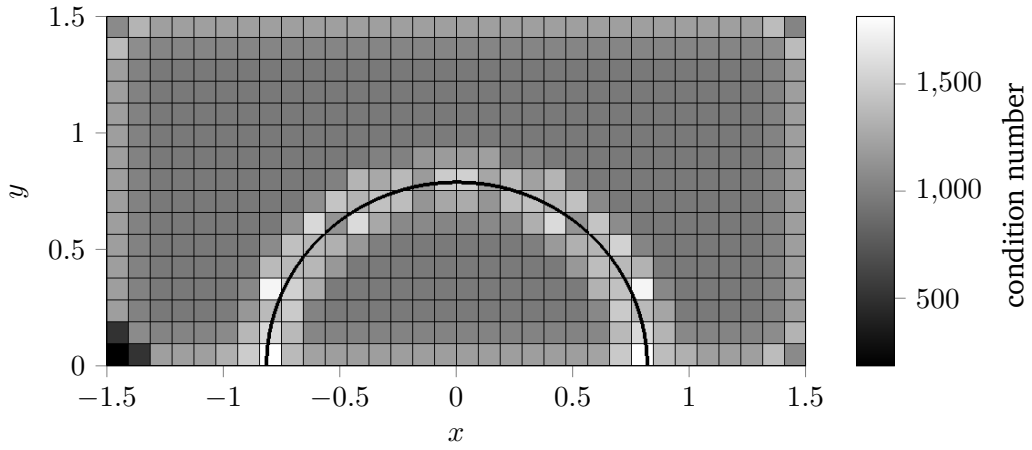


Figure A.4.: Stencil condition number on a 16×32 mesh for $\beta_S = 5$ and static contact angle of $\theta_{\text{stat}} = 120°$.

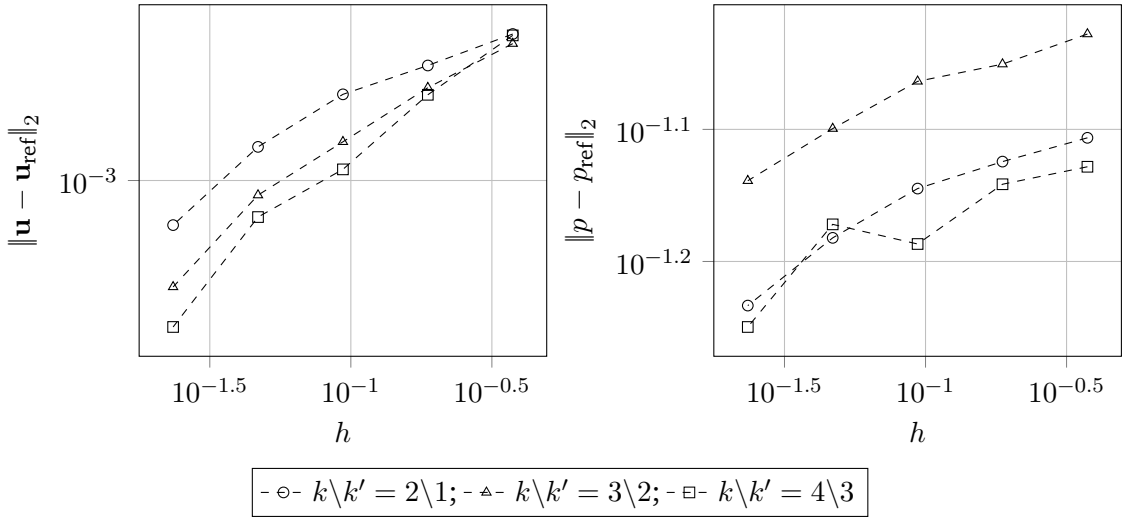


Figure A.5.: Mesh convergence study of a steady-state droplet with $\theta_{\text{stat}} = 120°$ on a slip wall with $\beta_S = 5$ (non equilibrium state) for $k = \{2, 3, 4\}$ on meshes with 4×8, 8×16, ..., 256×512 cells.

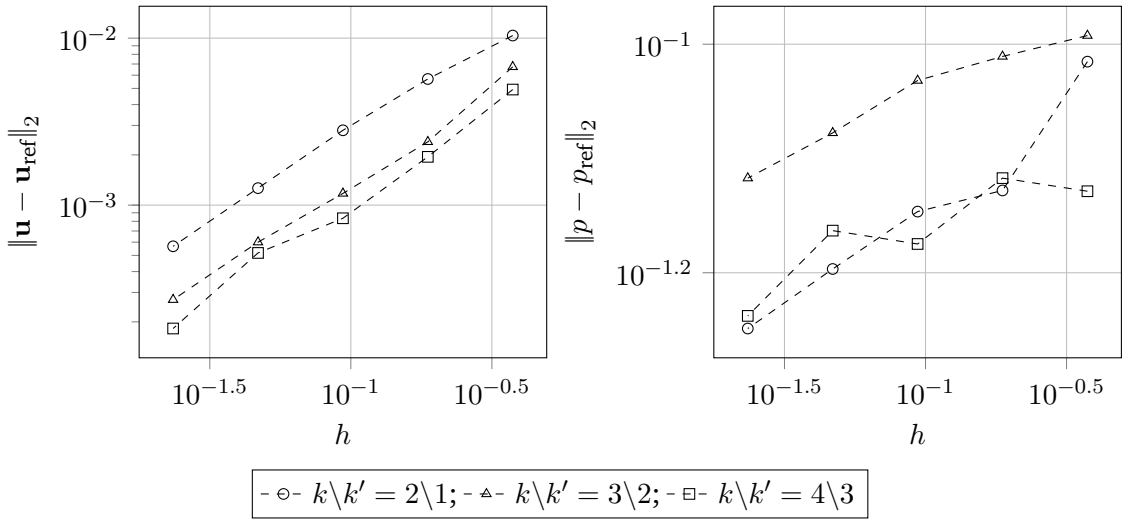


Figure A.6.: Mesh convergence study of a steady-state droplet with $\theta_{\text{stat}} = 120°$ on a free-slip wall with $\beta_S = 0$ (non equilibrium state) for $k = \{2, 3, 4\}$ on meshes with 4×8, 8×16, ..., 256×512 cells.

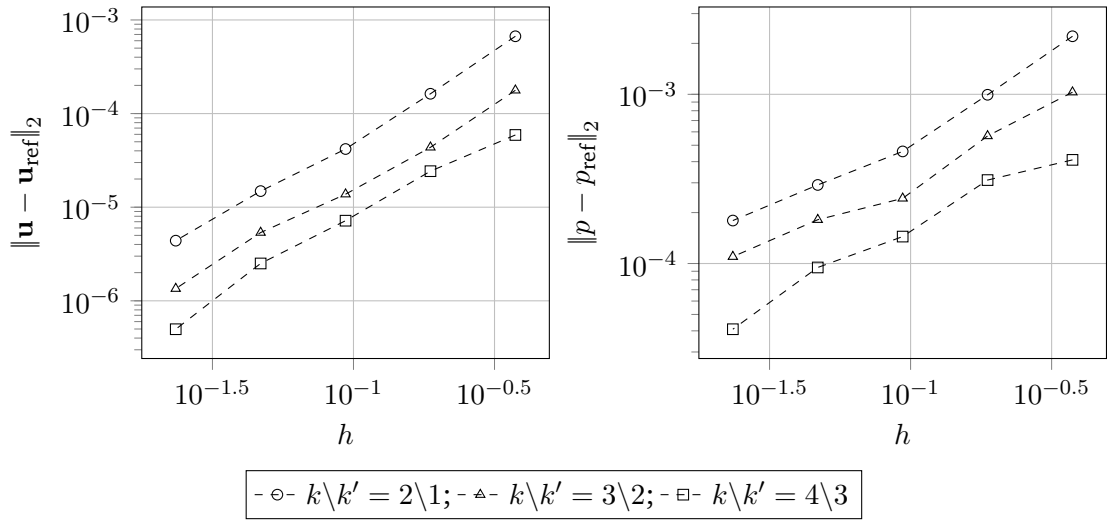


Figure A.7.: Mesh convergence study of a steady-state droplet with $\theta_{\text{stat}} = 90°$ on a slip wall with $\beta_S = 5$ (non equilibrium state) for $k = \{2, 3, 4\}$ on meshes with 4×8, 8×16, ..., 256×512 cells.

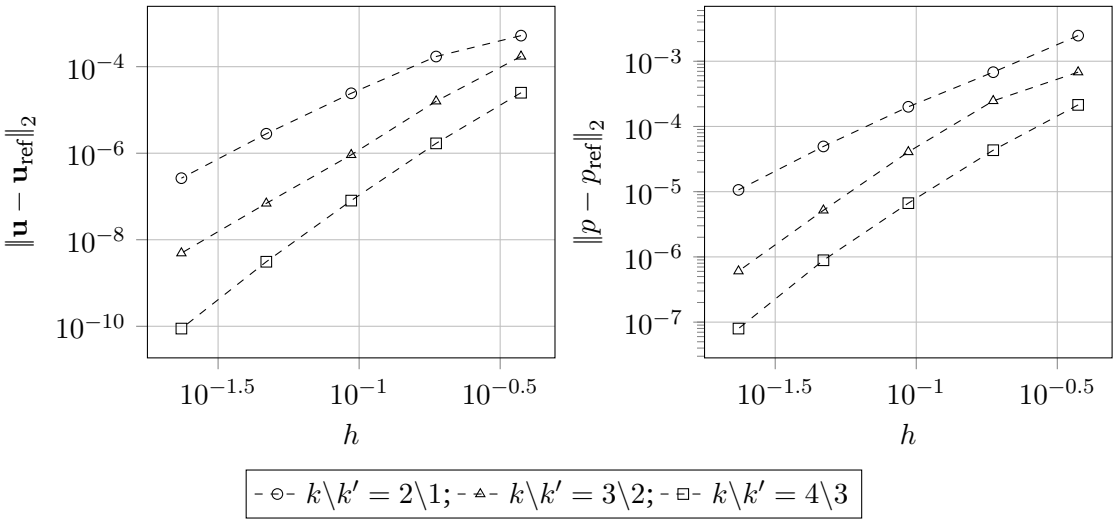


Figure A.8.: Mesh convergence study of a steady-state droplet with $\theta_{\text{stat}} = 90°$ on a free-slip wall with $\beta_S = 0$ (non equilibrium state) for $k = \{2, 3, 4\}$ on meshes with 4×8, 8×16, ..., 256×512 cells.